



City Research Online

City, University of London Institutional Repository

Citation: Stampoultzis, M. (2004). The use of XML schema and XSLT rules for product information personalization. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/8453/>

Link to published version:

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.



The Use of XML Schema and XSLT Rules for Product Information Personalization

by

Michael Stampoultzis

Research Supervisor

Dr. Bill Karakostas

A thesis submitted to

the City University of London

for the degree of

Doctor of Philosophy

School of Informatics

Centre of Human Computer Interaction Design

London

July 2004

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. Vassilis Karakostas. His domain knowledge, research experience and willingness guided me to the completeness of this thesis. His advises influenced personally me, and my research work.

DECLARATION

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

I grant powers of discretion to the University Library to allow this thesis to be copied in whole or in part without any reference to me. This permission covers only single copies made for study purposes, subject to normal condition of acknowledgement.

ABSTRACT

This thesis describes research carried out in order to help solve the problem of personalization in e-commerce/CRM system. Web-based personalization consists of activities, such as providing customised information, that tailor the user's Web experience- browsing a Web site or purchasing a product, for example-to that user's particular needs. The main research objective of the project is to investigate how XSLT technologies can be used for the development of matching engines that find XML represented products that match the tastes, needs or requirements of customers as captured in customer profiles, also represented in XML. More specifically our research investigates into novel algorithms for transforming XML based product specifications using rules that derive from mining customer profiles with the purpose of customizing the product information.

TABLE OF CONTENTS

THE USE OF XML SCHEMA AND XSLT RULES FOR PRODUCT INFORMATION PERSONALIZATION.....	1
TABLE OF CONTENTS.....	5
CHAPTER 1 INTRODUCTION.....	7
1.1 PROBLEM BACKGROUND	7
1.1.1 APPROACHES TO PERSONALISATION	8
1.1.2 SCHEMAS AND CUSTOMER PROFILES	14
1.1.3 RESEARCH ISSUES IN CONSTRUCTING CUSTOMER PROFILES.....	15
1.2 FOCUS OF THE THESIS	17
1.2.1 RESEARCH METHODOLOGY-OBJECTIVES	17
1.2.2 RESEARCH HYPOTHESES	18
1.2.3 RESEARCH OBJECTIVES	18
1.2.4 RESEARCH APPROACH.....	19
1.2.5 RESEARCH CONTEXT	20
1.2.6 NOVELTY OF THE DESCRIBED RESEARCH.....	21
1.3 STRUCTURE OF THE THESIS	22
1.4 CHAPTER CONCLUSIONS	24
CHAPTER 2 BACKGROUND AND STATE OF THE ART IN TECHNOLOGIES FOR PRODUCT PERSONALISATION.....	26
2.1 INTRODUCTION	26
2.2 DEFINITIONS OF PERSONALISATION	28
2.3 BUSINESS MOTIVATIONS FOR PERSONALIZATION	29
2.3.1 CUSTOMER EMPOWERMENT	29
2.3.2 CUSTOMER FOCUS.....	30
2.4 PERSONALIZATION AND CRM.....	31
2.5. ELECTRONIC CUSTOMER RELATIONSHIP MANAGEMENT (ECRM).....	32
2.6 PROBLEMS OF PERSONALIZATION	33
2.6.1 ASSESS DATA QUALITY.....	33
2.6.2 PRIVACY ISSUES	34
2.7 TECHNIQUES FOR PERSONALISATION	35
2.7.1 COLLECTING VISITOR INFORMATION	35
2.7.2 ANALYZING VISITOR PROFILES	38
DATA MINING TECHNIQUES FOR WEB BASED PERSONALISATION	47
<i>Data Mining</i>	48
<i>Web Mining</i>	48
2.7.3 MAKING RECOMMENDATIONS	51
2.8 XML TECHNOLOGIES FOR PERSONALISATION SYSTEMS.....	51
2.8.1 XML SCHEMA.....	51
2.8.2. XSLT	52
2.8.3 PRODUCT SCHEMAS	53
2.8.4 CRML.....	53
2.9 OTHER PERSONALIZATION APPROACHES	54
<i>Abstracting a Web site into a program</i>	54

2.10 CHAPTER SUMMARY 57

CHAPTER 3 ARCHITECTURE OF A XSLT BASED PRODUCT INFORMATION
CUSTOMISATION SYSTEM 58

3.1 INTRODUCTION 58

3.2 CONCEPTUAL ARCHITECTURE OF THE PROPOSED SYSTEM..... 60

3.3 REPRESENTATION OF CUSTOMER PROFILES AND PRODUCT/SERVICE MODELS
AS XML SCHEMAS..... 63

 CUSTOMER PROFILE..... 64

 PRODUCT SCHEMA TRANSFORMATIONS 65

3.4 ALGORITHM FOR CREATING XSLT PRODUCT FEATURE SELECTOR..... 68

3.5 WORKFLOW FOR CUSTOMER PRODUCT MATCHING SYSTEM..... 74

3.6 CONCLUSION..... 81

CHAPTER 4 DETAILED DESIGN AND IMPLEMENTATION OF THE PERSONALIZATION
SYSTEM 83

4.1. INTRODUCTION..... 83

Logical View..... 85

Deployment View..... 86

4.2. SYSTEM REALISATION MODELS 87

 4.2.1 *Use Case Model*..... 87

 4.2.2 *Analysis Model*..... 92

4.3 DESIGN MODEL 107

Create XSLT Rules Design Package (Subsystem)..... 108

Purchase Product Design package (Subsystem)..... 110

4.4 IMPLEMENTATION APPROACH..... 115

 4.4.1 SYSTEM DESCRIPTION 116

XSLT Rules Creator Application Description..... 116

4.5 CONCLUSIONS 152

CHAPTER 5 CONCLUSIONS 153

5.1 INTRODUCTION 153

5.2 REVISITING THE THESIS OBJECTIVES..... 156

5.3 RESULTS ACHIEVED 158

5.4 RESEARCH HYPOTHESES VALIDATION..... 159

5.5 FURTHER RESEARCH 161

5.6 CONCLUDING REMARKS..... 162

REFERENCES..... 163

CHAPTER 1 INTRODUCTION

1.1 Problem Background

As business-to-customer electronic market has already become an established way of doing business on the web, competition in the online retail marketplace is growing. Profit margins are affected from this stiff competition for existing competitors and also for new companies entering the market. To endure in this environment and gain a competitive advantage, a successful online vendor must provide a collection of captivating, personalized services that meet its customer needs. Gradually more, e-commerce sites are investing more on personalization studies in order to solve this problem. Internet's expansion now days makes more possible that most companies they will have their first interaction with a potential customer via the company's web site.

As a result, it is crucial for companies, to personalize their web sites, in order to match and transform web site contents, e-mail and other communication means, used to interact with customers, to individual customer preferences. Companies seeking to make potential customers increase the amount of time that they spend on their site, as well as increase their willingness to return to their site, which might lead to increased possibilities of purchasing a product. The company's objective is not to sell only a product or a service to a specific customer, but to increase the possibility that he/she is going to become a loyal customer and the company can eventually sell items and services to the specific customer over his/her lifetime. The ability to collect customer data exceeds our capability to analyze that data. And our capability to analyze customer data also exceeds our ability to act on that analysis while interacting with the customer. As a result, businesses lose the personal touch and push away their customers. Large amounts of unfiltered content and raw data are forbidding business from personalizing the customer experience, making customers to quit from interactions with the plain impression that the enterprise doesn't treat them as individuals. This leads to disappointment, lowers customer tendency toward loyalty and increases the probability that customers will try to find a different online retailer.

An obvious solution to this problem is the automation of the personalization process, i.e. the utilization of automated techniques that manipulate the vast masses of product and customer data. Personalization process involves three distinctive phases.

1.1.1 Approaches to Personalisation

The personalisation process involves the following steps:

Collecting visitor information. The objective of gathering customer information is to develop a profile which describes a set of descriptors important to the web site owner.

The techniques used the most are:

- **Explicit Profiling.** Every customer is asked to fill out information or online forms. “This method acquires user profiles by (1) asking users to answer preliminary questionnaires about topics or keywords which they are interested in, or (2) asking users to grade the pages they have browsed for interest and relevance.” [Hijikata, 2004].
- **Implicit profiling** monitors the visitor's behavior. The technique is generally not obvious to the visitor. Implicit profile construction is based on factors such as :
 - The time the user spends in order to read a page (browsing time) [Morita & Shinoda, 1994]
 - The detailed mouse button operation or the scroll operation while navigating through pages. [Sakagami & Kamba, 1997]
- **Using legacy data** accesses legacy data for important profile information, such as credit applications and previous purchases

Filtering the acquired visitor information. Four different automated techniques are described that manipulate the vast masses of product and customer data and facilitate the process of personalization.

- **Simple filtering technique.** Simple filtering relies on predefined groups, or classes, of users to determine what content is presented or what service is provided. An example of simple filtering is managing access to corporate information. For example, employees identified with the Human Resources department might have personalized Web sites that give them access to information and applications specific to their job.
- **Rules-based personalization** uses complex business rules, which depict business practices in logical constructs, i.e. in the form of a conditional statement: If X and Y, then Z. Rule-based techniques can be used in conjunction with filtering techniques, in order to develop the best recommendation. [Renneberg & Borghoff, 2003], “This

type of filtering method uses rules to transform the content of the customer profile into adjustments of the product model to the customer.”

A big advantage of rules-based solutions is that unlike click stream-based technologies, such as collaborative filtering engines and cookies, business rules can be used in any platform and therefore are not restricted to the Web.

Disadvantages of using business rules include:

According to [Pierrakos *et al.* 2003] rule based systems suffer from the same problems as other manually constructed complex systems, i.e., they require considerable effort in their construction and maintenance.

- a) Personalization business rules can get very complicated. A rule engine or a rule-base is needed in order to manage the rules. A rule management system is needed in order to test and validate new rules before they are applied in production. Controls are needed in order to ensure that a new rule won't duplicate or conflict with an existing rule. Producing effective CRM personalization business rules is very difficult and challenging, and it requires both business knowledge and technical skills.
- b) Rules-based personalization implies and requires that a detailed knowledge of what the customer wants. The solutions are also not very scalable, because the business rules need to be constantly altered and changed in order to capture the customer needs. A well-defined process for changing the rules is needed. Also users can play a vital role by modifying the rules by providing explicit preferences and commands.
- c) Business rules cannot scale to handle millions of customers on an individual basis. It is simply too complicated. To simplify the challenge, business rules typically are applied to customers that are grouped into segments with similar behavior patterns. Because of this, business rules cannot handle true one to one marketing, and do not deliver on the true value of decision making at the customer level.

Content-based filtering. This group of techniques applies machine learning methods to Web content, primarily text, in order to discover the personal preferences of a user. Content-based filtering works by analyzing the content of the objects to form a representation of the visitor's interests. “Content-based filtering systems are solely

based on individual users' preferences. The system tracks each user's behaviour and recommends items to them that are similar to items the user liked in the past". [Eirinaki & Vazirgiannis, 2003].

Disadvantages of using Content-based filtering:

According to [Pierrakos *et al.* 2003] "The main problem with content-based filtering is the difficulty of analyzing the content of Web pages and arriving at semantic similarities." There are different kinds of data represented in different ways in a web page. There isn't any unified way in order to represent data. This leads to confusion. Even if we ignore more complicated forms of data such as multimedia content, natural language itself is an amorphous source of data. Regardless of the considerable progression achieved in recent years regarding the analysis of textual data, there is no way of finding, in the near future, a way of making a program or a machine to understand natural language as humans do.

[Balabanovic & Shoham, 1997] also addressed the content-based filtering problem. They described a content limitation problem. According to them, "IR (Information retrieval) methods can only be applied to a few kinds of content, such as text and image, and the extracted features can only capture certain aspects of the content."

According to [Pazzani, 1999], there are two important sub problems involved in the construction of a content-based filtering system. "The first is finding a representation of documents. The second is to create a profile that allows for unseen documents to be recommended."

Content-based filtering is the simplest filtering approach that filters content by keywords, or string matching. The majority of Web search engines use content-based filtering, but they manage to collect only a small part of the indexable Web, less than 30 percent, according to [Lawrence & Lee Giles, 1998] and also users have to go through many results to find out relevant selections. According to [Florescu *et al.*, 1998] low coverage is achieved because the majority of Web pages are dynamically created and therefore they are not directly accessible via hyperlinks. Another reason is the lack of well defined conceptual models for Web information retrieval.

Collaborative filtering is a technology which looks at a user's behavior, relates it to the behavior of other users, and makes recommendations of other products or

information that the user will care about. Collaborating filtering system makes recommendations to a specific user which are based on other “similar” users who have purchased products before. What the system tries to do is to guess the personal preferences of an individual based upon the preferences of other users considered compatible with the individual. This approach makes the assumption that individual’s tastes are generally the same with the taste of another individual belonging to a group, or with the taste of a group of users of the system.

Advantages of using Collaborating Filtering

- a) The collaborative Filtering approach tries to solve the problem which the content-based filtering method failed to tackle. Collaboration Filtering based recommendation systems don’t use the content of the items for recommendation. Collaborative Filtering recommendations based on the assumption that if user A has similar interests to user(s) B interests, the items chosen by B can be suggested to A.

Disadvantages of using Collaborating Filtering

- a) They require considerable effort in order to be built and maintained.
- b) Most of the times collaborating filtering systems require the user’s involvement, which makes users to less keen to use the system.
- c) The nearest neighbour algorithm is the earliest Collaborating Filtering based technique used in recommendation systems [Resnick *et al.*, 1994]. With the use of this algorithm, the similarity between users is assessed based on their ratings of products, and the recommendation which is produced considers the items visited by nearest neighbours of the user. In its original form, the nearest neighbour algorithm uses a two dimensional user item matrix to represent the user profiles. This nearest neighbour algorithm has two different phases. As [Mobasher *et al.* 2001] says “there are two primary phases in collaborative filtering: the *neighbourhood formation* phase and the *recommendation* phase.” An algorithm used for the collaborative filtering is called (kNN) approach, or k-Nearest-Neighbour approach. This algorithm uses a user’s record of activity in order to compare that record with other historical records from different users in order to locate the top k users who might

have analogous preferences or interests. As we described above the correspondence of a visitor record to its *neighbourhood* is based on similarities based on:

- a) The way items are rated
- b) Access to pages containing similar content
- c) Purchasing similar products.

The selected neighbourhood is then used in the recommendation process in order to suggest items not already accessed or purchased by the active user.

Collaborative filtering-based techniques suffer from some limitations. Most of them are related to the scalability and efficiency of the kNN approach. And this is because kNN requires that the neighbourhood creation phase to be carried out as an online process, but for large volumes of data this may lead to unacceptable latency for providing suggestions.

[Shahabi & Chen, 2003], [Sarwar *et al.* 2000], [Zeng *et al.* 2004] also addressed the problems related to Collaborative filtering (CF). They identified the problems and they stated that most of them are originated from the usage of the kNN approach. The kNN approach causes the following problems:

1. Scalability: The execution time of the nearest-neighbour algorithm grows linearly and is related to the number of items and the number of users. As a result, the recommendation system cannot support large-scale applications with an efficient way because it takes too much time in order to traverse the matrix. Amazon.com provides more than 18 million sole items for more than 20 million people.
2. Sparsity: Because of the large number of items and user reluctance to rate the item, most often the profile matrix is incomplete. Consequently, the system cannot provide recommendations for some users, and the produced recommendations are inaccurate.

3. There is also another problem which is known as the *cold-start* problem [Huang et al. 2004]. The cold-start problem becomes obvious when a new user or item has just entered the system. Collaborative filtering cannot produce helpful recommendations for the new user because there are not so many previous ratings or purchases. In the same way, when a new item enters the system, it is most probable that collaborative filtering systems will not recommend it to many users because a small number of users have up till now rated or purchased this item. The cold-start problem can be seen as an individual case of the sparsity problem, in which case mainly the elements in certain rows or columns of the consumer-product interaction matrix CP are 0.

Different hybrid approaches have been developed in recent years trying to solve the problems mentioned above. Their objective was to combine and utilize different aspects and techniques used by the established personalization approaches in order to produce better results. [Smyth & Cotter, 2000] described a hybrid system called ClixSmart. ClixSmart employs two different content filtering strategies. A content based filtering approach recommends items that the user liked in the past. The collaborative filtering recommendation approach chooses items for a specified user that analogous users also liked.

Another hybrid system called Yoda [Shahabi & Chen, 2003], utilizes the advantages of clustering, content analysis, and Collaborate Filtering (CF) approaches.

We have taken a similar approach in order to construct a system which can offer some better results by combining different filtering techniques. Namely, it uses the concept of rules, as used by Rules-Based personalization systems. Its novelty though is that the rules are defined using a new standard (XSLT). Their maintenance and enhancement can be performed easily because XSLT standard represents data using tree structure (although we assume that rules creation is assisted by a data mining tool). This tree structure can be easily manipulated and changed by the use of document Object Model [W3C 2004]. By using XSLT we overcome the problem of maintenance and expandability that characterizes rules based filtering systems. Our approach shares also commonalities with collaborative systems, because such systems use extensively data mining tools, for knowledge extraction.

In order to overcome the content based filtering problem which is the inability of current techniques to parse and extract meaningful information from a web content we have used metadata. The XML standard and its components, namely XSLT, XPath were used extensively. By wrapping web content, such as text, pictures, and other content types residing

within a web page by using tags is an advance in a sense that we give to the wrapped element an identity. There are new standards [PDX] emerging which dictate the way by which items should be represented in order to facilitate their manipulation. Collaborative filtering technique is used in order to facilitate the creation of business rules and not to make suggestions. So we don't experience the sparsity and scalability problem.

We also overcome the problems of sparsity and scalability because we don't employ any kNN algorithms in order to produce recommendations for the current user.

1.1.2 Schemas and customer profiles

A Schema is a model which is used in order to describe the structure of information. Schema is a term taken from the database area in order to describe the structure of data in relational tables. In XML context, a schema describes a model. This model describes the possible arrangement of text and tags in a XML document. Schemas are used extensively to describe meta-data. They describe the structure of business documents. A *product schema* is a XML document which describes the structure of the product. The Product Data eXchange 1.0 standard [PDX] defines an XML encoding scheme that enables a total product definition.

Depending on the personalization methods used, there are different requirements for the contents and the representation of the Customer profile. Customer profiles are implemented as XML documents that are instances of XML Schema profiles. Examples of the concepts represented in a customer profile model include:

- Consumer Income.
- Social habits of the consumer
- Generally, factors that pertain to consumer behavior and reflect customer satisfaction and loyalty.
- Customer name, contact information, personal browser settings, address, payment information, IP-Address.

From a technical point of view, automatic personalization or recommendation means matching meta-information of products (stored in product schemas) against meta-information of customers (stored in the customer profile). The problem however is that product schemas do not always exist and customer profiles are often incomplete and defined in an ad-hoc way.

1.1.3 Research issues in constructing customer profiles

Customer needs is the driving force of the market. They change rapidly and this is why products have to be changed accordingly in order to fulfill and meet customer needs. Available product schemas become in short time obsolete. There is a continuous effort on trying to synchronize and keep up to date the product specification. Also customer profiles are not always accurate and trustworthy. There is also a problem with the reliability of the customer information. The basis of any personalization strategy relies on having robust and reliable customer data. The maintainability, accessibility and integrity of the customer data have to be accessed thoroughly. The quality and not the quantity of customer data are a decisive factor and a key element in the personalization process. Incoherent and conflicting customer data will ruin any personalization effort. Personal data included in the customer profile are usually stored in different storage media and in different formats. Keeping all these data synchronized and up to date is a potential problem for a large enterprise which deals with thousands or even millions of customers. Privacy issues also affecting the quality of customer profiles. One of the main reasons is the fears that many customers have about the way the enterprises manipulate and distribute their data. Most of the online users they don't want their private data to be manipulated in a vague way by enterprises. Most of them are quite negative when they are asked by companies to fill in online forms with information related to their personal data.

Recently in industries such as ship and aircraft building, where there is the need to manage large volumes of product data, formal product schemas have been introduced. Such schemas are defined in languages such as STEP [STEP]. STEP is an international standard (ISO 10303) for the representation of product data. The schemas defined in STEP are the result of many years of information modeling by industry experts in the various domains of product data. However, with the expansion of e-commerce and e-business product data need to be shared over the Internet.

Some times customer might want to do product comparisons, try to see if the product features are good enough for them and try to get the best offer and price. This is the case when the customer needs to access product information from a vendor site over the web. The internet has created a new landscape and a shift to customer power. The users now, are able to navigate an endless ocean of knowledge. The access to a big variety of information including product information and reviews about a specific product has made customers able to make more easily product comparisons and get the best deal. It is no longer necessary to go from store to store or through catalogues in order to compare products and prices, competitors are now only a click away. Customers are able to do everything from the comfort of their home.

Thus it is advantageous to have Internet based technologies that define formally product and customer schemas and manipulate customer data online on the Internet XML technologies can contribute to the solution of the problem of personalization

The advantages of using XML to construct product schemas are:

a. XML is a Meta language

XML can be used in order to define and describe any kind of information. By definition, XML is extensible. Standardization bodies, users and user groups use XML as a type of a platform-independent grammar in order to define mark up languages for specific purposes and schemas for all kinds of data models.

b. XML is ideal for structured documents

XML documents by definition are hierarchically structured. Nested XML document elements can be combined and nested in order to build complex information structures. In principle, it is possible to convert complex XML documents for use with relational technology, but this tends to be inefficient. Mapping XML elements to and from relational tables is a slow process and requires normalization and optimization effort before a XML document can be stored properly.

c. XML is open

XML standards are being supported by all major vendors. Sun Microsystems, Oracle, IBM, Software AG, Microsoft, have all seriously invested in XML and they are actively contributing until now in the standardization process.

d. XML is presentation neutral

Document presentation is separated from document content by the use of XML. How the documents are formatted and represented in a variety of devices is defined by the use of style sheets. Style sheets are applicable to any available XML document. A

piece of information can be presented in a personalized way by using XML and its associating standards, e.g. XSLT. The original document content still remains intact. Only its appearance is being affected by the XML standards. Some client-side reproduction of Web pages can be performed by using Extensible Style Language (XSL). Therefore some of the Web server's workload can be avoided.

Our research assumption therefore is that XML technologies such as XML schema and XSLT are suitable for handling complex product and customer profiles and can be used to realise product matching and recommendation systems

1.2 Focus of the thesis

This thesis describes research carried out in order to help solve the problem of personalization in e-commerce/CRM system. Web-based personalization consists of activities, such as providing customised information, that tailor the user's Web experience- browsing a Web site or purchasing a product, for example-to that user's particular needs.

1.2.1 Research Methodology-Objectives

This section clarifies the research objectives and describes the research methodology used to achieve these objectives.

More specifically our research addresses the following general research problems of personalization:

- Problem of understanding customer reasons for buying products as different customers have different motivations: To understand customer reasons we need to construct detailed and accurate buying profiles of customers that relate the customers' buying patterns and behavior to their gender socioeconomic (family status, income etc) and other relevant characteristics
- Problem of understanding and describing customer profiles: It is however very difficult to construct accurate and detailed customer profiles for reasons such as privacy and confidentiality.
- Problem of overwhelming available product information. During the last years the amount of information offered via the internet has grown incredibly. Google estimates, that its search engine has 3,083,324,652 websites [Neuhold, 2003]. It is obvious that a normal internet-user must have a personalized view of the topics which

are relevant to him. Not the quantity but the quality of the provided information determines whether a user is satisfied with a certain web-service or not.

- Problem that product specifications change frequently. This is something quite common to all kind of products and to all industries. Customer's appetite and preferences change quite fast and provided that customer needs are the driving force of the industry and the production we are facing this continuous alteration of product specification in order to meet customer needs.
- Problem of making customer interaction with the web site more intuitive and efficient. Constructing and tailoring a user interface which meets user's expectations and facilitates user's interaction with an enterprise is not a trivial task. Different web browsers for example, have different object model, different way of representing data to the user and are manipulated differently. There is not a unified way of presenting data to the user. Companies release interface construction guidance but they apply to their own product range only. We suffer from polyphony of different approaches on constructing intuitive and efficient user interfaces, personalized ones.

1.2.2 Research Hypotheses

The hypotheses made by this research are as follows:

1. XML technologies such as XML schema and XSLT can contribute to the automated product information personalization process.
2. Complex personalization rules can be defined using XSLT language. The basic assumption is that XSLT rules will be used to deliver more personalized and customized information to the customer.
3. Complex product models can be defined using XML schemas
4. Complex customer profiles can be defined using XML schemas

1.2.3 Research Objectives

The main research objective of the project is to investigate how XSLT technologies can be used for the development of matching engines that find XML represented products that match the tastes, needs or requirements of customers as captured in customer profiles, also represented in XML. More specifically our research investigates into novel algorithms for transforming XML based product specifications using rules that derive from mining customer profiles with the purpose of customizing the product information.

Other research objectives are:

- Research into techniques for constructing and representing customer profiles
- Research into concepts and techniques for product schemas
- Construct and validate algorithms for personalizing XML instances of product information using XSLT rules that match customer profiles to products

1.2.4 Research Approach

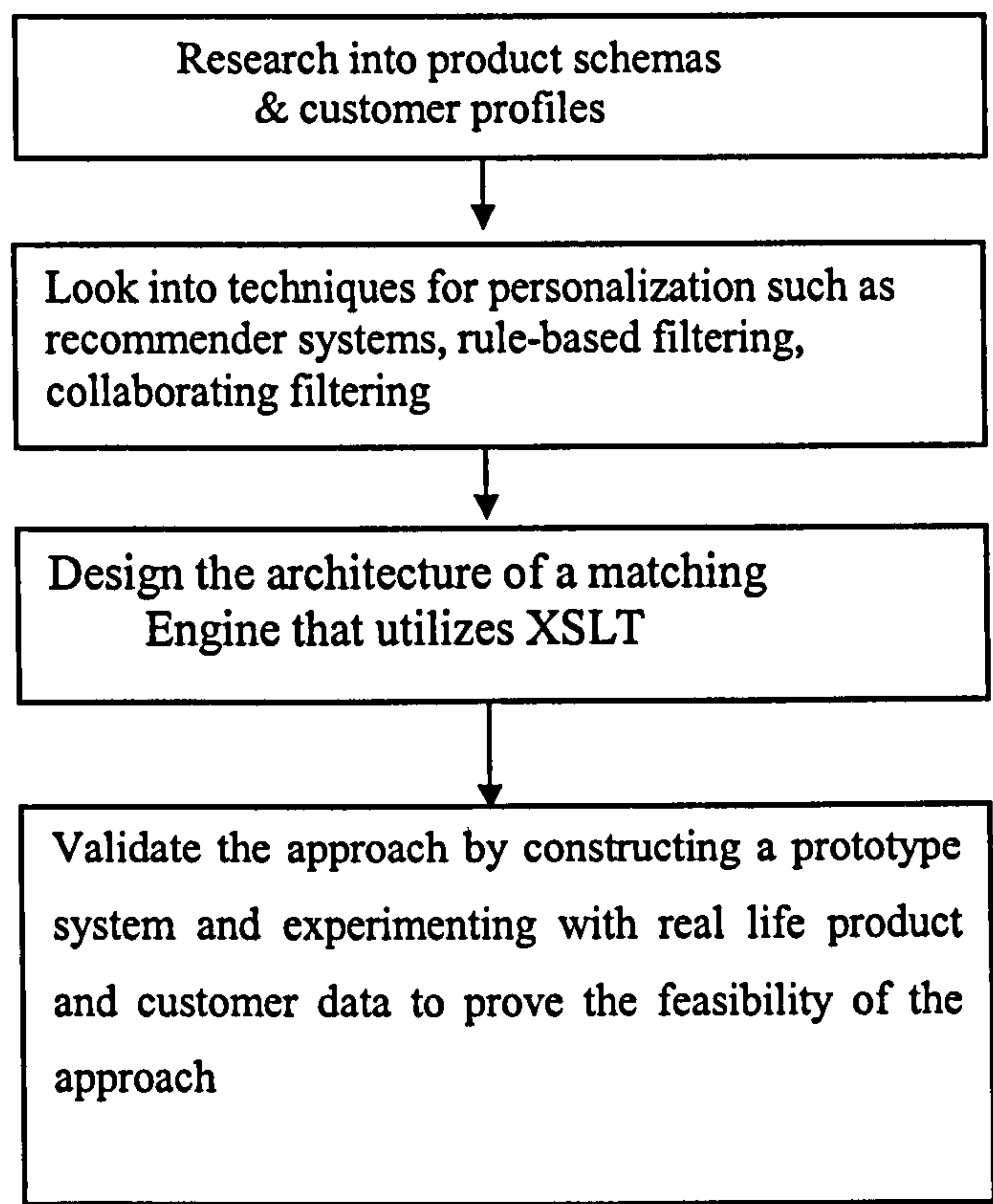


Figure 1: Research approach.

1.2.5 Research Context

POSITIONING OF OUR RESEARCH CONTEXT

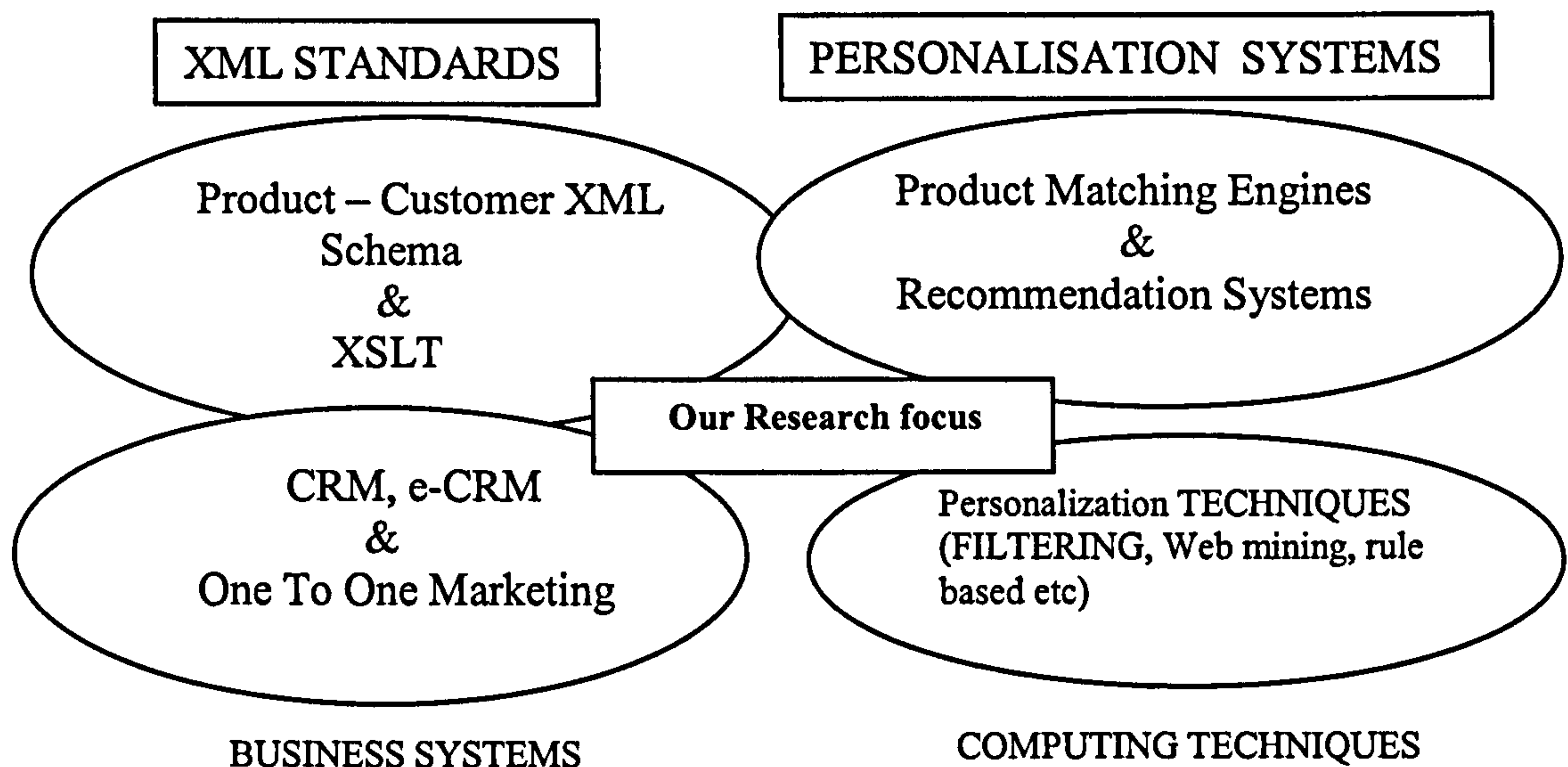


Figure 2: Positioning of research context.

The diagram in figure 2 depicts the context of our research. More specifically our research comes across the areas of:

- XML Standards. A XML Schema is a model which is used in order to describe the structure of information. XML Schemas used in order to create a blueprint, a prototype of our product and customer Meta data files. XSLT Is a language used to transform XML document into other XML document, or another type of document in web languages, such as HTML or XHTML. Its purpose in our research context is to be used in order to combine and link the Product and Customer Meta-data data and create a set of associative rules which are to be used in the personalization process.
- Business Systems.
 - a) CRM (Customer Relationship Management) systems are used in order to control efficiently the relationship between the customer and the enterprise. They become even more important because they automate customer-enterprise interaction and provide a business philosophy that involves

analysing, planning and controlling customer relationships by means of modern information and communication technologies.

- b) eCRM is a specialization of the CRM system. eCRM is a term often used to describe CRM conducted online.
- c) One to one marketing. Once they have gathered enough information about a potential client, either implicitly, e.g. by using cookies, or explicitly, e.g. by asking the customer to register online in their web site, the enterprises try to contact him by using different communication media, such as : email, phone call, post, customized pop-ups when he-she is browsing the enterprise's web site.
- Personalization techniques. Content-based filtering technique, rules-based filtering technique and collaborative filtering techniques and data mining covered. Data mining is the examination and analysis by automatic means, of large quantities of data in order to discover meaningful and significant patterns and rules. Enterprises use data mining in order to determine what their customers will do next. The association between data mining techniques and web based personalization is investigated. Web mining and its implications and contribution in the personalization process also considered.
- Personalization systems. Looking at the architecture of existing product matching engines. Investigate the kind of techniques that have been employed in order to perform product matching. Product matching engines are set of programs which deal with the task of associating a specific customer with a set of products that he/she might be interested on. The association is based on navigational patterns, previous purchases, purchases made by previous customers who have similar tastes to the current ones, also based to business rules that the company might have established and associated specific range of products to particular customer segments.

1.2.6 Novelty of the described research

Although many companies offer CRM products which give the ability to their clients to create Shopping Baskets they don't give the ability of personalization, and mass customization. The extraction and interpretation of a Data Mining Tool results and their representation as a viable, reliable and maintainable set of XSLT rules is one of the challenges. These XSLT rules actually are sets of personalization rules linking Product categories with Customer Segments. In this research we have assumed the existence of a data mining system that generates XSLT rules relating customer profiles to product attributes.

The introduction of metadata (XML standard), served a twofold purpose. The first one was to facilitate the creation of business documents which are compatible to the newest standards, they can be easily manipulated, sent easily through the internet by using current communication protocols. An open standard was used in order to solve problems related with data presentation and manipulation. The second one was to overcome the content based filtering problem which is the inability of current techniques to parse and extract meaningful information from web content. We achieved this by wrapping web content, mainly text and also pictures and some additional content types by using tags. These tags are used later in the process of finding and extracting useful and meaningful information out of the web content.

The XSLT [W3C 1999] standard, which is a subsequent part of the XML standard, is adopted for the creation of the association. This association is between products and customers. This association is expressed as a set of rules used by the rule-based system which is part of the hybrid system that we are constructing. Because of the XSLT nature, these rules can be represented as a tree. There is an already established technique of manipulating such kind of trees called DOM [W3C 2004]. Although rule-based personalization techniques suffer because the business rules need to be constantly altered and changed in order to capture the customer needs, the introduction of XML and XSLT and the usage of DOM (Document Object Model) make the task of alteration and synchronization and manipulation of business rules, embedded in an XSLT file, an easier task.

We are aiming in the construction of a hybrid system incorporating a combination of filtering techniques as it hasn't been tried before, namely rule-based and collaborative filtering. There are examples of hybrid system such as ClixSmart [Smyth & Cotter, 2000] which incorporates content based and collaborative filtering. Also Yoda [Shahabi & Chen, 2003], which utilizes the advantages of clustering, content analysis, and Collaborate Filtering (CF) approaches. Most of the hybrid systems use the content-based and collaborative filtering, filtering techniques combination.

1.3 Structure of the thesis

The thesis report is organised as 5 chapters.

Chapter 1 (this chapter) provides an overview of the problem background, namely the personalization problem. Also some well known personalization techniques are presented.

Some problems associated with the product and customer meta-data data then discussed. Then the focus of the thesis presented alongside with the research methodology, assumptions and research objectives of the thesis. The research approach and the novelty of the research follow. The structure of the thesis and the chapter conclusions complete the chapter.

Chapter 2 provides a Background and State of the Art in technologies for Product Personalisation. Formal definitions of personalization and web personalization are presented. The rational of using personalization is also presented. This section investigates the business motivation of using personalization. Customer empowerment and customer focus are selected and analyzed. Personalization problems and techniques are discussed in depth. Data mining and web mining are introduced and their role in web personalisation is discussed. Some emerging technologies which facilitate the personalization process are discussed here. More specifically XML Schema and XSLT are discussed. Also Product Schemas, which is a formal and widely accepted way of representing business document using Meta data, is discussed. CRML also introduced. A personalization approach which is using a set of Web sites as a program that abstracts the underlying organization of information is presented.

Chapter 3 chapter describes the conceptual and physical architectures of a system that matches customer characteristics to product information in order to present a personalized view of the product to the customer. A description of the system architecture at a conceptual level is given and also a brief description of the concepts on which the system is based. Some similarities and differences between the proposed personalization system and the known ones are discussed and some of the strong points of the system represented, such as: Connectivity, expandability, interaction abilities between different constituent components. It continues then by describing the Conceptual Architecture of the proposed system. It describes the overall architecture of the proposed system. It describes the interaction between the customer and the functional components of the system. It describes the different programming components which are working together in order to achieve personalization. Next, the representation of customer profiles and product/service models as XML schemas is covered. The requirements for the specification and modeling of product and customer information are represented. Also some examples with XSLT rules are given. These rules are produced by a Web Mining Tool which is a part of our proposed system. Some components of the system perform complex tasks, such as creating dynamically relationships and extraction rules for a specific customer and a product or range of products. The inner working of this component is discussed.

Chapter 4 describes the design and the implementation of the system. It begins with an overview of the application software which facilitates the creation of the design models. Lists the different models used in order to describe the overall architecture and design of the proposed demo system. Namely Use Case, Analysis, and Design model are represented. Also the Component view is included. The proposed demo system comprises two different set of programs. There is a description of these two different programs alongside with the description of the technologies which have been used in order to realize the proposed system. Then a demo is presented. There are two different users who create accounts in n online shop and then they decide to purchase a product. On purpose the scenario shows that they both buy the same product but the personalization process produces different results for the customers since they are having different customer profiles and different business rules are applied on these profiles in order to produce a personalized feeling in the interaction with the system.

Chapter 5 is the conclusion chapter. It starts by revisiting the notion of personalization. Brief summary of the available personalization techniques is given and also some of the problems associated with these techniques are presented. The thesis objectives revisited. The results of this work are then presented. After that a section which tests the research hypotheses presented. The chapter concludes with a section describing further research.

1.4 Chapter Conclusions

The main research contribution of this thesis is therefore a new method for managing complex product and customer information described as XML schemas and utilized with the use of XSLT rules in customer product match and recommender systems.

Essentially the research described in this thesis solves the problems that current rule based recommender systems suffer from i.e. The complexity of personalisation rules: By having these rules automatically generated the problem of updating product information structure and actual product data the personalisation rules in our approach act on well structured XML documents. The structure of these documents can be updated (e.g. new product characteristics added in the product description) but the existing personalisation rules will not be affected as long as the existing product information elements are not removed or changed. The problem of upgrading customer profiles. As the customer profiles are represented as XML schemas new elements can be added to the schemas. These will be ignored by the

existing personalisation rules. Potentially our approach can be scaled up to handling product and customer schemas with very large numbers of elements and with unlimited complexity. As the performance of XML technologies such as XSLT processors improves (for example by building XSLT processors) that access XML data from secondary storage like large relational databases) our approach can be scaled up to handle real life requirements. There are of course related areas of research that are not dealt by this thesis. These include Deriving personalisation rules by web mining. There are several research approaches that achieve that. These are discussed in chapter 2 of the thesis. The management (updating, deleting, consistency checking) of the automatically generated XSLT rules. The number of such rules can grow very large in proportion to the size of the product and customer schemas. How the proposed product customer match prototype is integrated with other personalisation/recommender systems. However as the prototype is based on open standards integration should not create much of a problem.

Chapter 2 Background and State of the Art in technologies for Product Personalisation

2.1 Introduction

This chapter investigates the state of the art research in number of related areas. These include:

- Definition of personalization. Different definitions taken from selected papers are presented.
- Web Personalization definition. Web Personalization is a special type of personalization which is applied on a web environment.
- The Rational of using personalization. This section investigates the business motivation of using personalization. Customer empowerment and customer focus are selected and analyzed.
- The relationship between Personalization and CRM. A CRM definition is included in this section.
- eCRM definition & usage. eCRM is a specialized version of CRM.
- Personalization problems.
 - ✓ Access Data Quality.
 - ✓ Privacy Issues.
- Personalization Techniques used in current systems. Personalization process involves three distinctive steps.
 - ✓ Collecting visitor information. The objective of gathering customer information is to develop a profile which describes a site customer's interests, purchases, or some other set of descriptors important to the web site owner. The techniques used the most are:
 - ❖ Explicit Profiling. Every customer is asked to fill out information or online forms. This method has an advantage because is allowing customers to describe to the site what they want to see.
 - ❖ Implicit profiling monitors the visitor's behavior. The technique is generally not obvious to the visitor. Browsing and buying patterns are the behaviors most often measured. The browsing pattern is monitored by using cookies.

- ❖ Using legacy data accesses legacy data for important profile information, such as credit applications and previous purchases. For existing customers and known visitors, legacy data often provides the richest source of profile information.
- ✓ Filtering the acquired visitor information. There are four different approaches that are presented here.
 - ❖ Rules-based filtering technique. Rules-based personalization uses complex business rules, which depict business practices in logical constructs, i.e. in the form of a conditional statement: If X and Y, then Z. Advantages and disadvantages for this specific technique are also presented. Some available systems which employ this technique are listed.
 - ❖ Simple filtering technique. Simple filtering relies on predefined groups, or classes, of users to determine what content is presented or what service is provided. An example of simple filtering is managing access to corporate information. For example, employees identified with the Human Resources department might have personalized Web sites that give them access to information and applications specific to their job.
 - ❖ Content-based filtering. This group of techniques applies machine learning methods to Web content, primarily text, in order to discover the personal preferences of a user. Content-based filtering works by analyzing the content of the objects to form a representation of the visitor's interests. Advantages and disadvantages for this specific technique are also presented. Some available systems which employ this technique are listed.
 - ❖ Collaborating filtering. Collaborative filtering is a technology which looks at a user's behavior, relates it to the behavior of other users, and makes recommendations of other products or information that the user will care about. Collaborating Filtering system makes recommendations to a specific user which are based on other "similar" users who have purchased products before. What the system tries to do is to guess the personal preferences of an individual based upon the preferences of other users considered compatible with the individual. This approach makes the assumption that individual's tastes are generally the same with the taste of another individual belonging to a group, or with the taste of a group

of users of the system. Advantages and disadvantages for this specific technique are also presented. Some available systems which employ this technique are listed.

✓ And finally making recommendations.

- Data Mining and Web Mining. Terms are explained. Their role in personalization and in web personalization process is discussed. Several research approaches in this area taken from selected papers are presented.
- XML Technologies for personalization systems. XML Schema and XSLT are defined. Product Schemas and CRML are defined.
- A hybrid personalization process is presented. There are different approaches for achieving personalization in a web site. Some are well known and established ones while others are emerging. Most of them are hybrid personalization methods and combine different characteristics and methods used by the well known ones. [Ramakrishnan, 2000] describes one of these hybrid personalization systems which is called PIPE (short for “Personalization is Part Evaluation”). PIPE employs a technique in order to personalize Web resources, without taking in account the previous interaction sequences. The input to a partial evaluator (which is a type of a compiler) is a program and its arguments. The output is a customized version of the program. The PIPE approach represents a set of Web sites as a program that abstracts the underlying organization of information. The program then with respect to the user input, is partially evaluated, and a tailored Web site is produced from the short cut version of the original program.

2.2 Definitions of personalisation

Several definitions of personalisation can be found in the literature.

According to [Tseng & Piller, 2003] “Personalization in general is about selecting or filtering information objects for an individual by using information about the individual (the customer profile) and then negotiating the selection with the individual. Thus, personalization compares strongly to recommendation”

In [Maurino & Fraternali, 2003] personalisation is defined as the process of gathering and storing information about the visitors of a Web site, analyzing the stored information, and, based on such analysis, delivering to each visitor the right information in the right way with the appropriate access rights.

Web personalization is defined as any action that adapts the information or services provided by a Web site to the needs of a user or a set of users, taking advantage of the knowledge gained from the users' navigational behavior and individual interests, in combination with the content and the structure of the Web site. [Eirinaki *et al*, 2003]

A web personalization system's purpose is to present users who interact with the company's Web site with related information they want or they might need, without the user having to ask for this kind of information explicitly. [Mulvenna *et al*, 2000]

Automatic personalization or recommendation can be achieved by matching meta-information of products against meta-information of customers represented as customer Xml schemas. Now days personalization is considered as an important component of Web applications. Techniques for personalization are being used in order to tailor information services to personal user needs. Also in marketing, personalization provides and facilitates one-to-one marketing. [Peppers & Rogers, 1997]

Another usage of personalization includes the so-called *recommender systems* [Schafer *et al*, 1999]. These systems have become very popular in the world of e-commerce.

Using automatic personalization, recommendations specific to customer are selected from a wide range of alternatives and represented to him. [Imhoff *et al*, 2001] [Elofson & Robinson, 1998] [Resnik & Varian, 1997].

Recommender systems "learn" from the customer and, amongst the large number of alternatives, offer recommendations that correspond closest to his/her user profile. The idea is to improve user access to relevant products and information by making personalized suggestions.

2.3 Business Motivations for Personalization

2.3.1 Customer Empowerment

The internet has created a new landscape and a shift to customer power. The users now, are able to navigate an endless ocean of knowledge. The access to a big variety of information including product information and reviews about a specific product has made customers able to make more easily product comparisons and get the best deal. It is no longer necessary to go

from store to store or through catalogues in order to compare products and prices, competitors are now only a click away. Customers are able to do everything from the comfort of their home.

Advancements and refinements in enterprise information systems made enterprises able to calculate the value of their customers and to realize the economics of customer churn. Enterprises have realized the high cost of replacing defecting customers with new ones. Enterprises realized the value of their existing customers, the fact that they were an important asset that needed to be managed carefully. They have turned their attention to issues like satisfying and retaining their existing customers, as well as identifying potentially profitable customers of the future.

2.3.2 Customer Focus

Since the introduction of the web and its wide adoption as the medium used to do business, many things changed in the way that companies use to promote and sell their products. Today companies tend to be more customer-centric. Personalization emerges as a key strategy in the creation of a customer centric enterprise. One of the objectives of companies is to respond to customer needs in real time. In a changing world, adapting to new trends and offering what customers need gives an advantage to the enterprise. The retail sector is the sector which is more customer-focused and customer-centric and the one which can evaluate and measure customer profitability and servicing costs. Service quality and interaction with the customer is a definite and crucial factor for long term sustainable success and development of today's enterprises. Customer focus is a critical success factor because customers are the source of income for the enterprises. Products and services which are tuned for individual customer result in customer loyalty which results in repeat business and increased profit.

Professor Sandra Vandermerwe [1997] suggests that in order to achieve customer loyalty and a sustained competitive advantage enterprises need to do things in a different way. A Customer-Centric business model needs to be adapted instead to the Product-Centric business model which is currently in use. To retain customers, enterprises need to be different from their competitors, in order for the enterprise to achieve customer 'lock-on', i.e. to make the customer choose to do business with the enterprise, because the enterprise understands her needs and provide them with what they need when they need it.

Previous techniques used by enterprises in order to 'lock-in' customers included charging customers a penalty fee for changing banks or switching mortgages. This wasn't a good way to create loyalty, but a good way to discourage customers from returning back as potential customers in the future. The chances of these customers to return back in the future were

limited and most of them they were leaving the enterprise whenever they were released from their contracts with the enterprise. With the customer centric business model the enterprise's knowledge of the customer acts as a fence to competitive entry.

Once a company manages to establish a relationship with a customer, then this relationship will continue to grow and increase in value. In contrast to product oriented companies where the value of the product fades away with time. The relationship between a customer and an enterprise increases in value as the time passes, because the longer the enterprise interacts with the customer, the more opportunities arise in order to learn about the customer changing needs and preferences. By knowing more about their customer needs and preferences they are able to tailor specific offer and services to meet these needs and more opportunities arise for revenue generating up selling and cross selling. The customer prediction model also gets better and the enterprise is making better predictions about future customer needs. This is why this type of relationship is described as a 'learning relationship'.

As [Hannigan & Peledrano, 2002] say, "customer-centric enterprises must be able to learn from each customer interaction and determine how they can best serve the customer over their lifetime with the enterprise". Effective personalization strategies increase customer switching costs and act as a barrier to competition which doesn't have the same amount of customer information as the enterprise does for its customers. As a result competitors can't provide the same level of personalized service to attract enterprise's existing customers. This advantage can't be matched easily by competitors and is quite desirable. But this level of personalization isn't easily achievable and requires many organizational changes and alterations in order to align the organizational objectives for building customer value.

2.4 Personalization and CRM

Lately, businesses are increasingly realizing the importance of knowing their customers better. E-businesses marketing efforts are focused on building lasting ties with customers through improved Customer Relationship Management.

According to [Bernett & Kuhn, 2002] "Customer Relationship Management (CRM) is a term for methodologies, processes, software, and systems that assist an enterprise in managing customer relationships in an organized and effective manner. In this context, the term "customer" can include suppliers, sales leads, employees, as well as paying customers. The goal of CRM is to optimize profitability, revenue, and customer satisfaction by organizing the enterprise's processes toward providing consistent high quality service to the customer".

In [Grabner-Kraeuter & Moedritscher, 2002] Customer Relationship Management is defined as “a customer-oriented business philosophy that involves analysing, planning and controlling customer relationships by means of modern information and communication technologies.”

Fayerman [2002] defines CRM as “an enterprise wide business strategy designed to optimize revenue and customer satisfaction by organizing the institution around customer segments. This strategy may represent a serious change in the existing organizational culture and behaviour. The resulting transformation allows organizations to more effectively select, attract, retain, and even grow customers.”

In order to perform an efficient CRM campaign, it is important to communicate with the customers, to understand customer behavior, to influence their behavior and to initiate campaigns, which are going to attract and retain customers. Enterprises, which employ CRM and Internet Marketing, though, have to create sophisticated models for information acquisition, information storage, and decision support. For example, customer segmentation models, predictive model for customer behavior, detection of key events that trigger behavioral changes and seeking important data from customers in order to conduct forecasting examples are all elements of decision support in the CRM area. The significance of these models is even more important because of the one to-one customization and one-to-one marketing methods that are used more and more to focus on the individual customer. Modeling research and literature is currently focused on event analyses, data warehousing, data mining, OLAP (On-Line Analytical Processing) and some other statistical modeling techniques. Along with personalization, Customer Relationship management (CRM), are regarded as business tools which will bring substantial business benefits, including improved customer retention, lower costs, higher sales, and eventually increased profitability. But CRM is vague and little understood, is complex and most of the times quite costly, and can engage a confusing array of technologies, which might need major architectural choices to be performed before the implementation phase. Confusion is one of the main problems.

2.5. Electronic Customer Relationship Management (eCRM)

As [Greenberg, 2001] says, “Electronic Customer Relationship Management or eCRM is a term often used to describe CRM conducted online”. There is much argument as to whether eCRM is a valid description or just another term to describe the same old CRM. “...the difference between CRM and eCRM is mostly in the name” [Sweiger *et al*, 2002].

For the time being, though, eCRM should be considered independently to the traditional client/server CRM architecture. Most of the enterprises today tend to deal with their online

and off line customer interactions differently and for that reason tend to have different strategies. eCRM involves in its interaction with customers, media and system architectures that have created new opportunities and drawbacks, leading to new challenges about whether the customer relationships which are created across these interaction channels are strong enough. It has been proposed that enterprises should not abandon their traditional channels of communication (CRM) while trying to optimize the web-enabled channels. [Greenberg, 2001].

2.6 Problems of personalization

Companies, wanting to improve Customer Relationships with personalization, are required to take specific actions before making any implementation efforts. Although these actions might be seen as obvious ones, still many companies are facing problems that prevent them from realizing the full potential and benefits of a personalization strategy. Below are some preliminary actions that have the greatest impact on the project success.

2.6.1 Assess Data Quality.

According to [NISS, 2001], “Data are a product”, and “As a product, data have quality, resulting from the processes by which data are generated”.

Without data, which is consistent, reliable and easily accessible, moving towards personalization will be useless. There is a problem with the reliability of the customer information. The basis of any personalization strategy relies on having robust and reliable customer data. The maintainability, accessibility and integrity of the customer data have to be tested and accessed thoroughly early in the process of designing the personalization component. The quality and not the quantity of customer data are a decisive factor and a key element in the personalization process. Incoherent and conflicting customer data will ruin any personalization effort. It is quite important not to underrate the importance of the quality and integrity of the company’s data when a personalization system is being designed. Costly CRM projects failed to deliver due to issues with data reliability.

Forrester Research [Forrester, 2001] estimates the cost of a typical CRM program implementation between \$60 and \$130 million. Before moving to the implementation phase, the quality of customer data have to be accessed in order to determine if it can be trusted enough in order to achieve the desired personalization strategy.

If data are corrupted or can't be trusted enough a plan must be created in order to increase data reliability. Another problem is associated to the storage media, which hold the data. Product data, customer data, transactional data, they are scattered in different systems and they might be in different and diverse format. If the enterprise has more than one customer communication channel then this is the most probable scenario. For example, if the enterprise uses email campaigns, selling through call centers. Accessibility, maintainability and integrity of each data source have to be tested and checked. In case that there are multiple channels of communication between the customer and the company then the same piece of information about a specific customer might be stored in multiple systems. For example a customer's phone number. It might be stored in the sales database as well as in the call centre database. A way has to be found in order to determine which version of each piece of overlapping data is the most reliable and trusted. This has to be done continuously because data quality fades over time.

2.6.2 Privacy Issues

Personalization process involves a lot of customer data analysis and manipulation. But is getting more difficult, for companies, to acquire valuable and trusted customer data. One of the main reasons is the fears that many customers have about the way the enterprises manipulate and distribute their data. Most of the online users they don't want their private data to be manipulated in a vague way by enterprises. Most of them are quite negative when they are asked by companies to fill in online forms with information related to their personal data. According to [Kobsa, 2002] :“The privacy awareness of web users has grown significantly: They often leave web sites that require registration and even renounce shopping online due to privacy concerns”. This is a big problem for enterprises dealing with customization because personalization is based on reliable customer data. Companies have to find out what kind of customer data is relevant to the personalization policy and which part of the data might be privacy critical and act accordingly in order to protect that data. It's obvious that personalization and privacy don't mix. Protection of customer privacy is a big problem as more organizations and enterprises gather information about customers for personalization purposes. Companies with well established privacy policies were trying to sell personal information have produced more fears about customer privacy. Technical measures have to be

employed in order to guarantee a maximum of privacy for the customers. It is crucial for companies that collect information about customers to set up and follow a privacy policy and enforce company's policy validation by privacy certification organizations such as TRUSTe (<http://www.truste.org>) and BBBonline (<http://www.bbbonline.org/>).

2.7 Techniques for personalisation

Personalization of Web content is one of the fastest-growing segments of the Internet economy. Because it reduces Information overload and give users a customized experience whenever they are navigating a Web site, personalization has created a multimillion-dollar industry. There are numerous companies such as Netperceptions (<http://www.netperceptions.com>) and Yodlee (<http://www.yodlee.com>) which are building customized personalization solutions for individual client specifications. Personalization process involves three distinctive steps.

- Collecting visitor information.
- Filtering the acquired visitor information.
- And finally making recommendations.

2.7.1 Collecting visitor information

The objective of gathering customer information is to develop a profile which describes a site customer's interests, purchases, or some other set of descriptors important to the web site owner. The techniques used the most are:

- **Explicit Profiling.** Every customer is asked to fill out information or online forms. This method has an advantage because is allowing customers to describe to the site what they want to see. The user has direct control about the information which is to be displayed in the screen. In this environment, the user is able to directly manipulate both the kind of content seen on the page as well as its layout by choosing from a series of different options. For this reason, it has also been called "*check-box personalization*" [Mulvenna *et al*, 2000]. This type of site has been used a lot by Website portals that also interested in offering news to its customers. An example is MyYahoo, where the customer is asked to identify profile information, for example what sports he/she likes, what kind of magazines and newspapers, and what kind of

news categories to display. Then MyYahoo creates the web page based on customer's selection. Another example is MyExcite.com. By offering customization options, these sites hope to tempt users by offering them a virtual space where the right kind of information is laid out just the way they like it. In a nutshell, the hope is to give users the power to decide for themselves what they like.

Generally the steps that the user can take to personalize its page are the following:

- 1. Select personalization link from portal home Page.

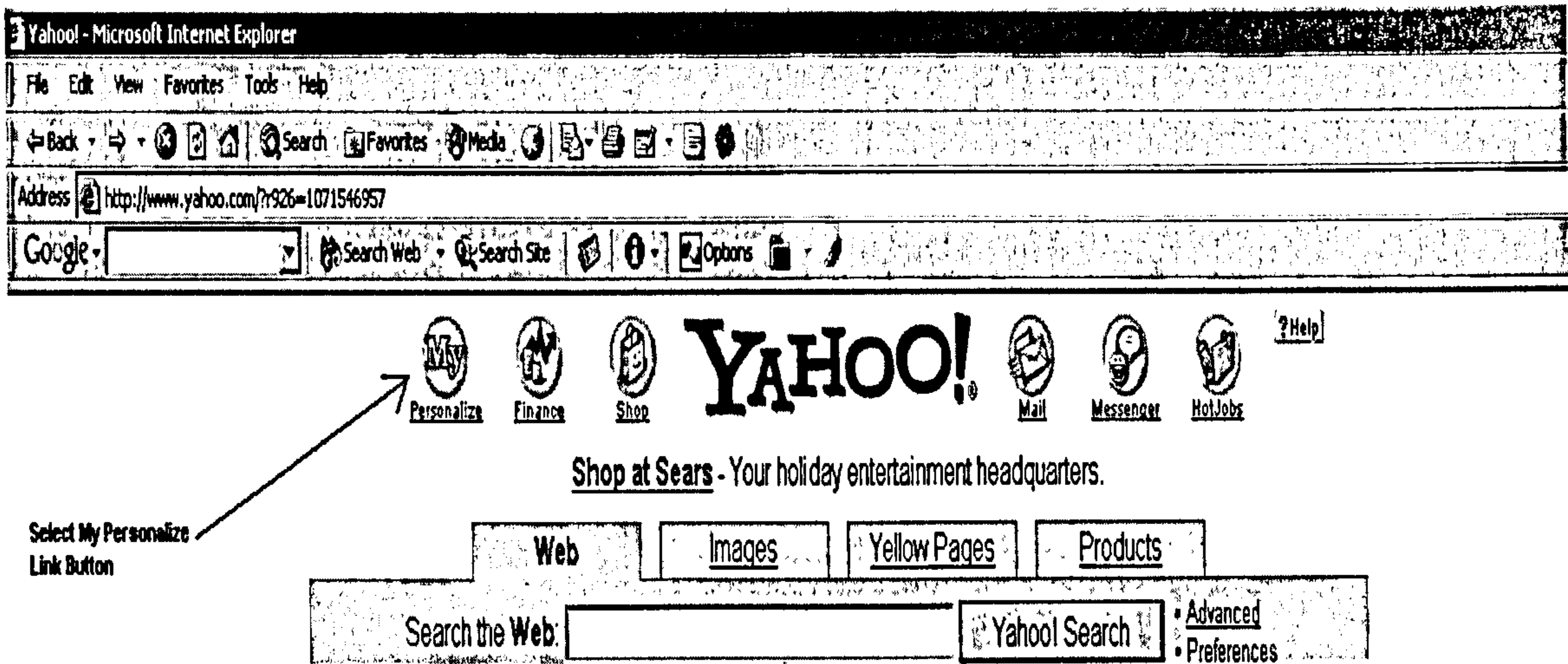


Figure 1: Yahoo Main Page.

- 2. Register your information
- 3. Choose content, color and layout options

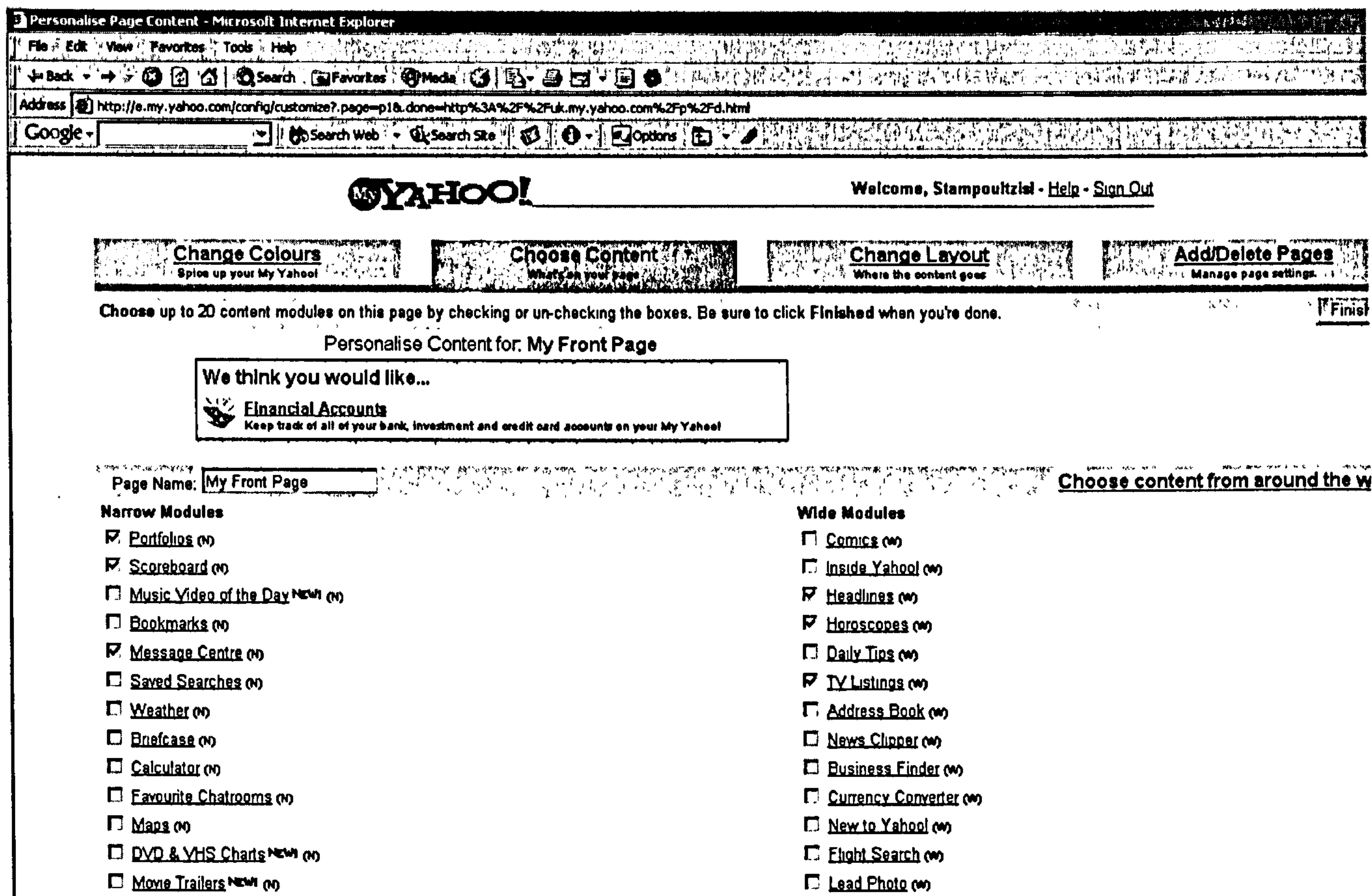


Figure 2: Content, Color and Layout Page.

Here the user is represented with a set of check box to select from. That is why this method is called Check Box Personalization.

4. Set as home page

- **Implicit profiling** monitors the visitor's behavior. The technique is generally not obvious to the visitor. Browsing and buying patterns are the behaviors most often measured. The browsing pattern is monitored by using cookies. Cookies are small piece of information, stored either in a file, or in the computer's memory and facilitates the visitor identification and behavior tracking. The buying pattern can be found by looking in the customer purchase database. For example, Amazon.com logs each customer's buying history and, based on that history, recommends specific purchases.
- **Using legacy data** accesses legacy data for important profile information, such as credit applications and previous purchases. For existing customers and known visitors, legacy data often provides the richest source of profile information.

2.7.2 Analyzing visitor profiles

After the profile is obtained, the next step is to process the profile information in order to display or make a recommendation about documents, purchases or some other type of information which is relevant with the customer's needs and preferences. Making these recommendations is the most difficult task. There are a number of techniques for presenting content and making suggestions and recommendations. Rule-based and filtering techniques are the ones which are used the most.

- **Rule-based filtering technique**

Personalization rules enable business or web administrators to specify which content to display on a Web page, based on user individuality or business conditions. A rule maps users to the target content. Rule-based techniques offer a visual editing environment for the business administrator to specify business rules in order to produce personalization. Most of the times alongside the business administrator, another person, a consultant, with knowledge of the business sector is consulted in order to produce the appropriate business rules. The rule-based approach provides a flexible mechanism to specify rules for business applications or marketing campaigns.

Rules-based personalization uses complex business rules, which depict business practices in logical constructs, i.e. in the form of a conditional statement: If X and Y, then Z. For example, a rule can be: If (customer is between 30 and 40) and (salary > £100,000), then deliver expensive content. Rules-based personalization engines can:

- Produce personalization content on the fly.
- Facilitate the cross-selling of relevant products. For example, a rule could be created in order to make a suggestion for a product A to a customer who has just purchased product B; for example, a customer of a book might be interested in current or previous books by the same author or in books on the same subject.
- Offer important advice, problem resolution tips, or both.

Rule-based techniques can be used in conjunction with filtering techniques, in order to develop the best recommendation. It's most successfully used with complex transactions, in which the vendor has clear, planned, and well-defined practices in identifying which is the customer target group it wants to address, and how it needs to

conduct these relationships. Rules-based personalization can be used to automate lengthy processes.

Advantages of using business rules:

- a) A big advantage of rules-based solutions is that unlike click stream-based technologies, such as collaborative filtering engines and cookies, business rules can be used in any platform and therefore are not restricted to the Web.

Disadvantages of using business rules:

- a) Personalization business rules can get very complicated. A rule engine or a rule-base is needed in order to manage the rules. A rule management system is needed in order to test and validate new rules before they are applied in production. Controls are needed in order to ensure that a new rule won't duplicate or conflict with an existing rule. Producing effective CRM personalization business rules is very difficult and challenging, and it requires both business knowledge and technical skills.
- b) Rules-based personalization implies and requires that a detailed knowledge of what the customer wants. The solutions are also not very scalable, because the business rules need to be constantly altered and changed in order to capture the customer needs. A well-defined process for changing the rules is needed. Also users can play a vital role by modifying the rules by providing explicit preferences and commands.
- c) Business Rules cannot scale to handle millions of customers on an individual basis. It is simply too complicated. To simplify the challenge, business rules typically are applied to customers that are grouped into segments with similar behavior patterns. Because of this, business rules cannot handle true one to one marketing, and do not deliver on the true value of decision making at the customer level.

Available Rules-Based Personalization systems

WebSphere, a system produced by IBM (<http://www-306.ibm.com/software/inf01/websphere/index.jsp>), has a set of tools and services which are used in order to enable an e-business development team to create a personalized web site. Personalization makes the site more attractive and easier to use, so attracts a larger audience and improves the service.

BroadVision Personalization Solutions, a system produced by BroadVision (<http://www.broadvision.com>), allows the organization to gather profile information, session and event-based observations, and transaction information in real time in order to help understand the web site visitors. Rule-based personalization allows to dynamically tailoring relevant information for different audiences in order to deliver a different and more personalized web and wireless experience.

E.piphany Real-Time, a system produced by E.piphany (<http://www.epiphany.com>), provides a strong, real-time marketing engine to give a detailed view of the customer and to personalize each interaction. Through a Web user-interface, it gives the ability to an administrator, to combine campaigns and offers across all customer touch points. The system employs a powerful combination of rules-based and analytic personalization technologies to deliver optimal marketing decisions for every customer relationship.

Filtering Techniques

Filtering techniques use algorithms to analyze Meta data and drive presentation and recommendations. The three most common filtering techniques:

- Simple filtering.
- Content-based filtering.
- And collaborative filtering

Systems which use Content-based filtering or collaborating filtering are called *Recommender Systems*.

Simple Filtering

Simple filtering relies on predefined groups, or classes, of users to determine what content is presented or what service is provided. An example of simple filtering is managing access to corporate information. For example, employees identified with the Human Resources department might have personalized Web sites that give them access to information and applications specific to their job. Online brokerages often classify their accounts by asset value or age groups. Their sites can use simple filtering in order to provide special treatment for customers based on whether they are different customer category. Or, when an age group is specified, the site could recommend savings accounts for retirement or some other expenditure.

Content-based filtering

As [Balabanovic & Shoham, 1997] say, “We consider a pure content-based recommendation system to be one in which recommendations are made for a user based solely on a profile build up by analyzing the content of items which that user has rated in the past.” Content-based approaches directly exploit the product information.

“*Content-based filtering* systems are solely based on individual users’ preferences. The system tracks each user’s behaviour and recommends items to them that are similar to items the user liked in the past”. [Eirinaki & Vazirgiannis, 2003]

Content-based use keywords, string matching, link patterns in order to provide simple “Web query languages” for personalization. Examples include WebSQL, WebOQL, and Florid. [Florescu *et al*, 1998]

This group of techniques applies machine learning methods to Web content, primarily text, in order to discover the personal preferences of a user. Content-based filtering works by analyzing the content of the objects to form a representation of the visitor's interests. Generally, the analysis needs to identify a set of key attributes for each object and then fill in the attribute values. One example is a document filtering system that analyzes documents based on keywords. Recommending video movie purchases is another example of content-based filtering. Content-based filtering is most suitable when the objects are easily analyzed by computer and the visitor's decision about object suitability is not subjective.

The makers of content-based filtering system argue that it is possible to uniquely characterize each patron without having to match his or her interests to another. This is because items are recommended based on information about the item itself rather than on

the preferences of the users of the system. In this way, it is possible to successfully recommend items that have not already been previously rated by other users. [Mooney&Roy, 2000] evaluated the performance of such a system, called LIBRA, and found that it was able to provide quality recommendations. They emphasize that a content-based recommender system works best for unpopular items for users with unique tastes; but effective content information about the items is uncomplicated. In LIBRA system each user when is registered is presented with a set of options to choose from. These options are represented as Radio Buttons.

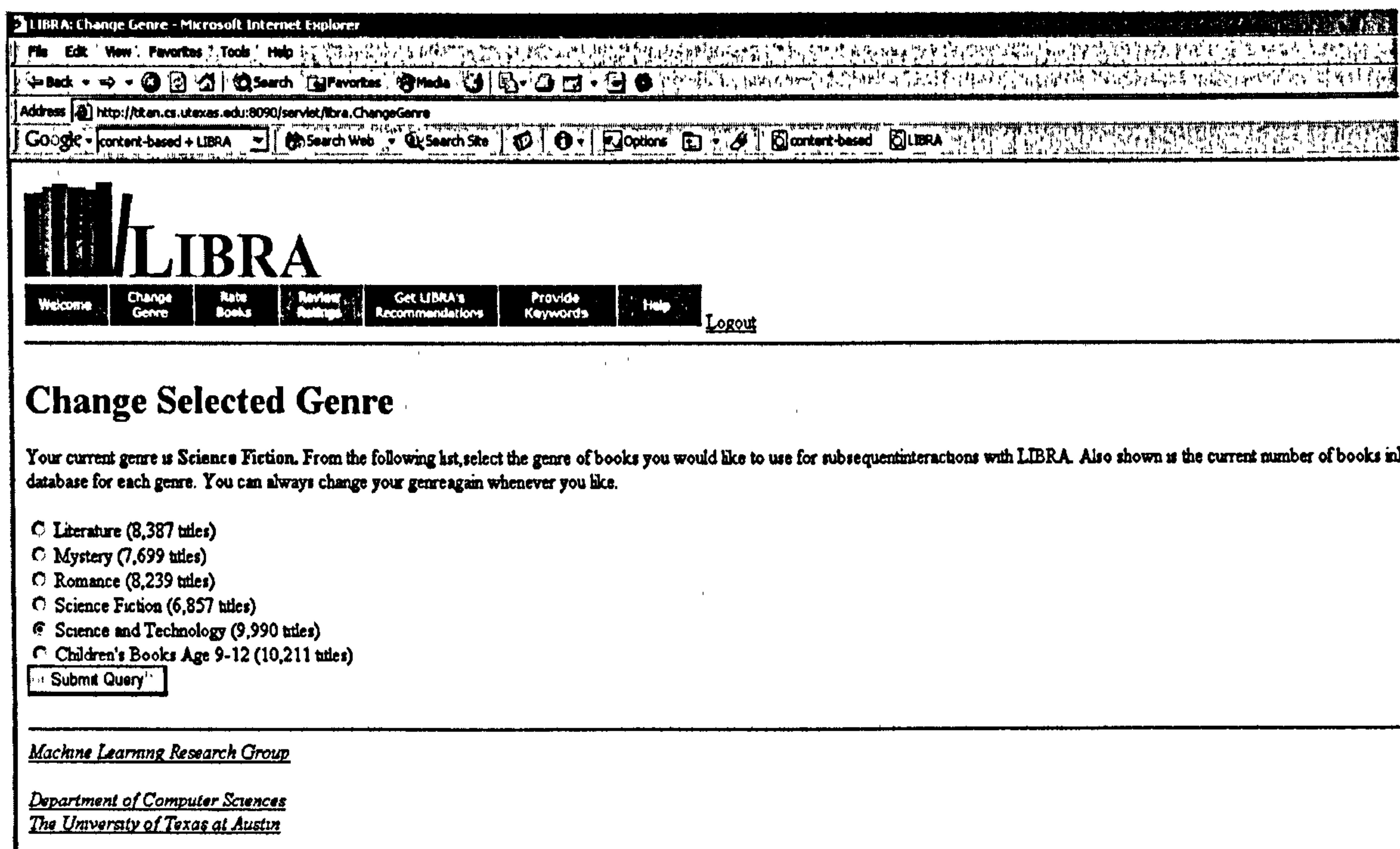


Figure 3: LIBRA System

After the user makes some selection in different screens then is represented with some suggestions.

Disadvantages of using Content-based filtering:

- **Content Limitation:** Information Retrieval methods can only be applied on a specific kind of content, such as text and image, and the extracted features can only capture certain aspects of the content. When a product consists of hundreds or even thousands of attributes then it is difficult for this approach to work efficiently, because content-based filtering is heavily based on previous user selection. Is most unlikely that a user will deal with this amount of information and therefore the feature recommendations which will be based on a partially finished selection process will be inaccurate.

- Content-based filtering is the simplest filtering approach that filters content by keywords, or string matching. The majority of Web search engines use content-based filtering, but they manage to collect only a small part of the indexable Web, less than 30 percent, according to [Lawrence & Lee Giles, 1998] and also users have to go through many results to find out relevant selections. According to [Florescu *et al*, 1998] low coverage is achieved because the majority of Web pages are dynamically created and therefore they are not directly accessible via hyperlinks. Another reason is the lack of well defined conceptual models for Web information retrieval.

Other examples of content-based filtering systems include:

InfoFinder, “is an agent that learns to find and categorize information using sample documents selected by the user. By using these samples, it extracts semantically significant phrases from documents and tries to learn optimal search queries for each category”. [Krulwich & Burkey, 1997]

NewsWeeder , [Lang, 1995], which is able to adaptively construct user models from a user's browsing behaviour, based on the similarity between Web documents containing news items. These models can be used to filter news items according to each user's requirements.

Collaborating Filtering

Social or Collaborative filtering (also called group filtering), gathers visitors' opinions about a set of objects, in order to form similar groups and then learns from the groups to predict a particular visitor's interest in an item. Instead of finding objects similar to those a visitor liked in the past, as in content-based filtering, collaborative filtering develops recommendations by finding visitors with similar tastes. Recommendations produced by collaborative filtering are based on the group's response and are not restricted to a simple profile matching. For product recommendations, collaborative filtering is most appropriate for identical, simple products, such as books, CDs, or videos.

Collaborative filtering is a technology which looks at a user's behavior, relates it to the behavior of other users, and makes recommendations of other products or information that the user will care about. Collaborating Filtering system makes recommendations to a specific user which are based on other “similar” users who have purchased products before. What the system tries to do, is to guess the personal preferences of an individual based upon the preferences of other users considered compatible with the individual. This approach makes the assumption that individual's tastes are generally the same with the

taste of another individual belonging to a group, or with the taste of a group of users of the system.

This technology compares information and identifies behavioral patterns by using a simple analysis of data relationships. This personalization technology operates on the assumption that groups of users share similar tastes; therefore if a customer is interested in a product X, he is going to be interested on a product Y, which many product X buyers have also purchased. This assumption isn't always true for many product and service categories, but it has been successfully when was used for books and movies.

Collaborative filtering is an effective recommendation engine, when it uses gathered preferences from groups of users. Is based on expressed preferences, either explicitly acquired, from data entered in online forms, responses to online surveys, or implicitly, by keeping track, with the use of cookies, of the pages a user visits and of the product(s) they buy. Collaborative filtering provides enhanced and more adaptive personalization. The weakness of this approach is that, it results in only vague classification of customers based on their preferences for specific products; it doesn't take into consideration the reason of the purchase. . And if the context is not factored e.g. if the customer bought a product for himself or it was destined for a present, the personalization results can be confusing.

Known Collaborating Filtering systems

- Collaborating filtering utilizes data mining techniques to provide information and recommendations based on a user's pattern of behavior over time. Firefly (<http://www.firefly.net>), which was acquired by Microsoft, is probably the best-known example of this approach.
- Another example of a system which uses collaborative filtering is the Amazon site (<http://www.amazon.com>). Whenever a customer buys a product then the site makes another set of recommendation-s based on previous users' purchases who bought similar products.

**TEXT BOUND
INTO
THE SPINE**

Shop now

Apparel & Accessories

by Jeffrey Richter (Author)



Price: \$34.99 & This item ships for FREE with Super Saver Shipping. See details.

Availability: Usually ships within 24 hours

Edition: Paperback

READY TO BUY

 **Add to Shopping**

**Sign in to turn on 1-
ordering.**

MORE BUYING CHO

18 used & new from
\$25.00

Have one to sell? Sell yo

Add to Wish List

Add to Wedding Registry

Don't have one?
We'll set one up for

► [See more product details](#)

Best Value

Buy Applied Microsoft .NET Framework Programming and get Programming In the .NET Environment at an additional 5% off Amazon.com's ever low price.



Buy Together Today: \$64.91  **Buy both and save**

[Click here for promotion rules](#)

Customers who bought this book also bought:

Here the system makes some additional recommendations. This is an example of collaborative filtering

- Programming Windows with C# (Core Reference) by Charles Petzold (Author), Charles Petzold (Hardcover)
- Programming Microsoft .NET by Jeff Prosise (Author), Jeff Prosise (Hardcover)
- Microsoft ADO.NET (Core Reference) by David Sceppa (Author), David Sceppa (Hardcover)
- Programming C# by Jesse Liberty (Paperback)
- Advanced .NET Remoting by Ingo Rammer (Paperback)

► **Explore Similar Items:** 20 in Books, 1 in Magazine Subscriptions, and 1 in Software

Figure 4: Amazon.com. Example of collaborating filtering

- Another example is PHOAKS at AT&T Labs [Terveen *et al.*, 1997]. Users send their opinions and rates about web resources in Usenet Netnews. PHOAKS parses, categorizes and abstracts those opinions automatically.

Advantages of using Collaborating Filtering

- The collaborative Filtering approach tries to solve the problem which the content-based filtering method failed to tackle. Collaboration Filtering based recommendation systems don't use the content of the items for recommendation. Collaborative Filtering recommendations based on the assumption that if user A has similar interests to user(s) B interests, the items chosen by B can be suggested to A.

Disadvantages of using Collaborating Filtering

- They require considerable effort in order to be built and maintained.
- Most of the times collaborating filtering systems require the user's involvement, which makes users to less keen to use the system.
- Users who are first rating new items they won't be given recommendations at all.
- The nearest neighbour algorithm is the earliest Collaborating Filtering based technique used in recommendation systems [Resnick *et al*, 1994]. With the use of this algorithm, the similarity between users is assessed based on their ratings of products, and the recommendation which is produced considers the items visited by nearest neighbours of the user. In its original form, the nearest neighbour algorithm uses a two dimensional user item matrix to represent the user profiles. This version of collaboration filtering-based recommendation system has two problems:

1. Scalability: The time complexity of executing the nearest-neighbour algorithm grows linearly with the number of items and the number of users. Thus, the recommendation system cannot support large-scale applications with an efficient way because it takes too much time in order to traverse the matrix. An example is Amazon.com, which provides more than 18 million unique items for over 20 million users.

2. Because of the large number of items and user reluctance to rate the item, most often the profile matrix is incomplete. Consequently, the system cannot provide recommendations for some users, and the generated recommendations are inaccurate.

None of the available personalization techniques can be applied on every web site and none of them can satisfy all the personalization goals. Different personalization techniques are suitable for different things. Selling books or CDs, it involves different techniques than the ones which have to be employed in order to sell computers, clothes, groceries. An enhancement could be the creation of a technique which improves the current ones, and also offers additional options which can satisfy a wider set of needs by employing a flexible architecture which allows the accommodation and cooperation of multiple recommendation engines, and each one of these engines can use a specific personalization technique to make the recommendations.

Data mining techniques for Web based personalisation

Behavior analysis which can be achieved by using algorithms, samplings, parallelisms and other techniques to discover patterns and predict possible behavior is still an ongoing process with a lot of problems. Consumers don't readily fall, most of the times, into the categories that statisticians create for them, and also their behavior can't be predicted by using observations based on the past. The problem is to discover what customers want at the moment they surf the internet and prepare themselves to buy something, something which can't be determined by only analyzing the past.

Data mining techniques achieve:

- **Customer segmentation groupings** (dividing customers into those most and least likely to repurchase a product).
- **Profitability analysis** (which customers created the most profit for a specific period of time).
- **Personalization** (the ability to market to individual customers based on the data collected about them).
- **Event monitoring** (e.g. when a customer reaches a predefined level of purchases); what if scenarios (E.g. what's the possibility for a customer or a customer category who bought one product to buy an analogous one).
- **Predictive modeling** (E.g. comparing a range of product development plans in terms of possible future success, given the customer knowledge base). Data collection and analysis is a continuing and iterative process. It is assumed that business decisions are refined over time, based on feedback from earlier analysis and decisions.

Potential benefits of data mining techniques lead to better and more productive customer relations in terms of sales and service, as well as better supply chain management (lower inventory and speedier delivery), resulting in lower spending and more competitive pricing. One of the major challenges is how to integrate the data mining techniques with existing legacy systems.

Data Mining

According to [Ester *et al*, 1999] “Data mining is the step of applying appropriate algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data.”

Data mining is the examination and analysis by automatic means, of large quantities of data in order to discover meaningful and significant patterns and rules. Enterprises use data mining in order to determine what their customers will do next. Predictive modeling aids companies in order to predict what one of their users will do next. Multiple actions by the user are considered in determining the eventual outcome.

Data mining applications use different methods including decision trees, neural networks and other combinations of business rules to target a specific customer. These state-of-the-art methods include sophisticated statistical tools such CHAID (Chi-Squared Automated Interaction Detector, a market segmentation technique) and CART (classification and regression trees); smart data mining applications that use algorithms such as Bayesian networks to analyze and predict demand; and online analytical processing engines to crunch the numbers.

Web Mining

According to [Kosala & Blockeel, 2000] “Web Mining is the use of data mining techniques to automatically discover and extract information from web documents and services”.

There are several research approaches in this area. Most of these research approaches are concentrated on extracting useful patterns and rules using data mining techniques in order to comprehend the users’ navigational behaviour, and then consequently all tasks concerning site reorganization or modification can be made by humans. In numerous cases, a recommendation engine assists the customer’s navigation through a site. Enhanced

functionality is provided by some other more advanced systems. The notion of adaptive web site is introduced and means for changing dynamically the site's structure are provided. These research efforts combine more than one method in Web personalization, specifically, user profiling, content management, Web usage mining techniques, and publishing techniques.

Some of the most interesting research approaches are listed bellow:

One of the first attempts to make use of the information that can be obtained through exploring a customer's navigation through a Web site resulted in the creation of a system called Letizia [Lieberman, 1995], which is a client-site agent which monitors the customer's browsing behavior and seeking for potentially interesting pages for recommendations. The client-site agent looks at the pages residing in the neighbor of the active Web page and by using a best-first search enhanced by heuristics tries to guess user's interest and offer some suggestions.

[Shahabi et al., 1997] suggested the use of a client-site agent which monitors the client's behavior and creates a customer profile. Then the system creates groups of users with similar interests. The design and implementation of a profiler which is used to capture the client's selected links and page order is discussed and explained. Profiler also gathers information about the page time viewing and cash references, and all these are done by using a Java based remote agent. All the information accumulated by the profiler is then utilized by a knowledge discovery technique which groups users with similar interests. The novelty of this approach is based on the fact, that a new path clustering method based on the similarity of the history of user navigation is introduced. This approach is able to capture user's interests which could persist through a number of consequent hyper text link selections. A simulation is being performed for the evaluation of their path clustering technique by using a sample WWW site. The result showed that they can recover the correct clusters by 10%-27% of average error margin.

Cooley et al. [1999b] and Srivastava et al. [2000] define Web usage mining as a three- steps process, comprising of pre-processing, pattern discovery and pattern analysis. Their suggested system, called WebSIFT, initially performs cleansing and pre-processing for identifying the users, the server sessions, and also gathers cashed page references by using a special field called the referrer field and also performs content pre-processing [Cooley et al. 1999a]. General statistic algorithms alongside

data mining techniques such as association rules, clustering, sequential pattern analysis, and classification have been employed for the purpose of pattern discovery. At the end the results are analyzed by using a simple knowledge query mechanism, a visualization tool, or an information filter, which makes usage of the pre-processed content in order to automatically filter the results of the knowledge discovery algorithms.

Masseglia et al. [1999a, b] uses Web mining techniques such as sequential pattern discovery and association rules on Web log files. The acquired knowledge then used in order to modify the server hypertext organization dynamically. In their approach Web usage mining is regarded as a two-phase process, consisting of the pre-processing phase where all unrelated data are removed and the log file entries are grouped and this grouping is based on time considerations, and the Web mining phase in which data mining techniques are applied. The system that they suggest called WebTool, additionally provides a query language used to enhance the mining process. A different component producing dynamic links based on the rules generated from sequential patterns or association rules, and whenever a visitor's navigation pattern matches a rule, the hypertext organization is changed dynamically.

Another system proposed by Mobasher et al. [1999, 2000a] is called WebPersonalized and is regarded the most sophisticated one. The proposed system provides a framework for mining Web log files in order to find out information for the preparation of recommendations to customers based on their browsing patterns to previous users. The system is depended on anonymous usage data provided by logs and the structure of the site. When the data assembly and the pre-processing phase ends, data mining techniques such as sequential pattern discovery, association rules, classification and clustering are applied, in order to find out useful usage patterns. The results from this phase are used for the creation of aggregated usage profiles, which are used for the creation of decision rules. A recommendation engine associates user's activity against these aggregated profiles and proposes to him/her a list of recommended hyperlinks. Some new enhancements have been added recently [Mobasher et al. 2000b,c] in order to incorporate content profiles into the recommendation process. Content profiles and usage are described as weighted collections of page view records. Content profiles identify

alternative ways in which pages with similar content can be grouped. The system contains two different modules: the offline, which includes data preparation and some specific Web mining tasks, and the online component, which is a real-time engine producing recommendations.

2.7.3 Making recommendations

- Once the first two initial steps are finished, namely collecting visitor information and filtering the acquired visitor information, the last step is to make some recommendations

2.8 XML technologies for personalisation systems

2.8.1 XML Schema

XML documents can be read by humans, and they are text representation of data. These XML documents need a well defined structure in order to be portable across different platforms and systems used for development. A way to achieve this is to develop a XML Schema. The purpose of a XML Schema is to describe a class of XML Documents using XML mark up constraints in order to document the usage and the relationship of elements, data types and content. XML Schemas resemble classes and XML Documents are like objects. This is analogous to object oriented concepts. Normally a class is like a template and is being used in order to instantiate the objects. XML documents are referred as instances and XML schemas are classes.

A Schema is a model which is used in order to describe the structure of information. Schema is a term taken from the database area in order to describe the structure of data in relational tables. In XML context, a schema describes a model. This model describes the possible arrangement of text and tags in a XML document. A schema can be considered also as an agreement on a commonly accepted vocabulary for a specific application which involves the exchange of documents.

“An XML Schema is a set of schema components (Schema component is the generic term for the building blocks that comprise the abstract data model of the schema). There are 13 kinds

of component in all, falling into two groups. The primary components, which may (type definitions) or must (element and attribute declarations) have names are as follows:

- Simple type definitions
- Complex type definitions
- Attribute declarations
- Element declarations

The secondary components, which must have names, are as follows:

- Attribute group definitions
- Identity-constraint definitions
- Model group definitions
- Notation declarations “

[W3C 2001]

2.8.2. XSLT

XSLT Is a language used to transform XML document into other XML document, or another type of document in web languages, such as HTML or XHTML. XSLT which stands for Extensible Style sheet Language Transformations is part of XSL (Extensible Style sheet Language). XSL consist of three parts:

- XSLT
- XPath which is also a language for defining parts of a XML document. XPath is used extensively by XSLT.
- XSL-FO is a language for formatting XML documents

The main functionality of XSLT is the transformation of one XML document to another, and also the addition or removal of new elements into the output file. It can sort or rearrange the XML file elements, make decisions based on conditions which elements to display. XSLT uses XPath in order to define matching patterns for the purpose of transformation. XSLT became a Web Standard by a W3C (World Wide Web Consortium, www.w3c.org) recommendation in 16th November of 1999.

XSLT Definition

“A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source tree. A template is instantiated to create part of the result tree. The result tree is separate from the source tree. The structure of the result tree can be completely different from the structure of the source tree. In constructing the result tree, elements from the source tree can be filtered and reordered, and arbitrary structure can be added.”

[W3C 1999]

2.8.3 Product schemas

The Product Data eXchange 1.0 standard defines an XML encoding scheme that enables a total product definition to be described at a level appropriate to facilitate supply chain interactions. The scheme is defined for bill of materials (BOM), approved manufacturer list (AML), changes (Engineering, Manufacturing, Product) and references to documents describing geometric and other definition characteristics. Standard transfers the data required to support the following business processes: (1) Quote request; (2) Manufacturing; (3) Engineering change management, including signoff; (4) Work in Process; (5) Report on Quality; (6) Report on as-built product configuration. [PDX]

2.8.4 CRML

CRML provides an open, application- and vendor-independent method for describing any type of customer relationship [CRML]. The OASIS CIQ Technical Committee plans to adopt CRML into its family of customer information specifications including xCIL and xNAL, which define unique customer characteristics.

"Up until now, the OASIS CIQ Technical Committee has concentrated on defining standards to describe characteristics of a customer, such as name, address, phone number, email, etc. Now with CRML, it is possible to describe relationships between customers from a business and personal point of view," said. Under the CRML standard, a customer can be a person or an organization. CRML accommodates complex customer relationships including person-to-person, person-to-business, and business to business.

2.9 Other Personalization Approaches

There are different approaches for achieving personalization in a web site. Some are well known and established ones while others are emerging. Most of them are hybrid personalization methods and combine different characteristics and methods used by the well known ones. [Ramakrishnan, 2000] describes one of these hybrid personalization systems which is called PIPE (short for “Personalization is Part Evaluation”). PIPE employs a technique in order to personalize Web resources, without taking in account the previous interaction sequences. The input to a partial evaluator (which is a type of a compiler) is a program and its arguments. The output is a customized version of the program. Partial evaluation is traditionally used to make a program faster and to remove interpretation operating cost. Is a technique which facilitates the simplification of program presentation by discarding pieces of code implementing the business logic (which in this particular instance is the customization or personalization logic) that doesn’t apply to a specific customer. The PIPE approach represents a set of Web sites as a program that abstracts the underlying organization of information. The program then with respect to the user input, is partially evaluated, and a tailored Web site is produced from the short cut version of the original program.

Abstracting a Web site into a program

Consider the structure of a Web site, which is organized in a hierarchical way. The web site is abstracted as a tree representation. In Figure 5, the nodes (rectangles) represent individual Web pages for the site and the links represent some tagging mechanism, such as the HTML (an <href> that associates hyperlinks in a Web page). This model assumes that the web site is a structured one and its structure is well defined. In this case a parsing program employing a depth first search can be used in order to produce a program which models the web pages themselves and the links between them and includes all the navigational scenarios. For example, the data in Figure x corresponds to the following program:

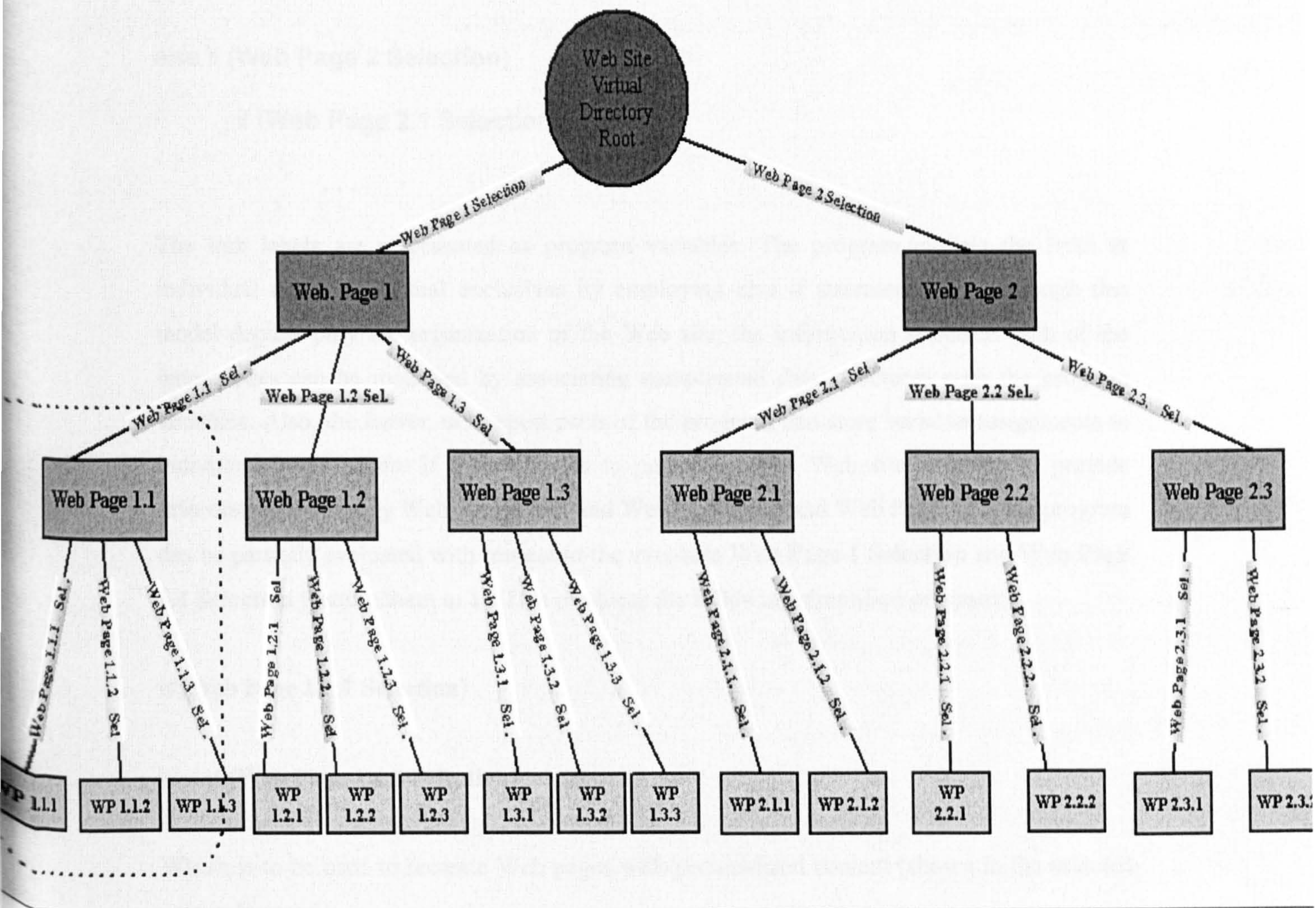


Figure 5 First three levels of a hierarchically organized Web site. Lower levels could include detailed information. The labels on edges represent selections made by a user. The circled region indicates the personalization output.

if (**Web Page 1 Selection**)

 if (**Web Page 1.1 Selection**)

 if (**Web Page 1.1.1 Selection**)

 ...

 else if (**Web Page 1.1.3 Selection**)

 ...

 else if (**Web Page 1.2 Selection**)

 ...

...

else if (**Web Page 2 Selection**)

 if (**Web Page 2.1 Selection**)

 ...

The link labels are represented as program variables. The program models the links at individual nodes as mutual exclusives by employing else if statements. Even though this model depicts only the organization of the Web site, the information stored at each of the inner nodes can be modelled by associating complicated data structures with the program variables. Also, the *leaves*, or deepest parts of the program, can store variable assignments to individual home pages. If a user wants to personalize the Web site in order to provide information about only Web Page 1.1.1. and Web Page 1.1.2 and Web Page 1.1.3 the program can be partially evaluated with respect to the variables **Web Page 1 Selection** and **Web Page 1.1 Selection** (setting them to 1). This produces the following simplified program:

if (**Web Page 1.1.1 Selection**)

 ...

else if (**Web Page 1.1.3 Selection**)

 ...

Which is to be used to recreate Web pages with personalized content (shown in the selected area in Figure 5).

This approach demonstrates the concept of using partial evaluation for personalization, but it is only working properly if there is a well defined structure in the site. Most Web sites are not austere hierarchical and several use links for reason other than narrowing on an information resource. But this idea can be used in order to address the selection of specific properties of products once these products are represented by using Meta Data description files. By nature these Meta Data files are hierarchical and can be represented like trees. The personalization process in the selection of specific elements out of the products can be achieved not by anticipating each user's selection and accordingly reducing the size and structure of the program which abstracts the site but by the means of another Meta Data file which describes the preferences of the specific user.

2.10 Chapter Summary

This chapter investigated the background and the state of the art in technologies used for product personalization. More specifically the following areas were covered: Personalization and its definition were introduced. Web personalization was also introduced. Web personalization is a special type of personalization applied on a web environment. The Rational of using personalization and the business motivation for personalization were discussed. CRM was introduced. Its relationship with personalization analyzed. A specialized version of CRM, ECRM was introduced. Some of the personalization problems were introduced. More specifically Access Data Quality and Privacy Issues were discussed. Personalization Techniques used in current systems were also discussed. The distinctive steps involved in personalization process were discussed. Data Mining and Web Mining terms were explained. Their role in personalization and in web personalization process was discussed. Several research approaches in this area taken from selected papers were presented. Web mining is assumed to be a prerequisite to our approach. We assume that the results of data mining are input to our system in term so XSLT rules and being processed by a XSLT-based Product Information Customization System. XML Technologies for personalization systems presented. XML Schema and XSLT were defined. Product Schemas and CRML were defined. Also a hybrid personalization technique called PIPE was presented. The PIPE approach represents a set of Web sites as a program that abstracts the underlying organization of information. The program then with respect to the user input, is partially evaluated, and a tailored Web site is produced from the short cut version of the original program. In the next chapter we will represent the conceptual and physical architectures of a system that matches customer characteristics to product information in order to present a personalized view of the product to the customer. A brief description of the concepts on which the system is based is given. Some similarities and differences between the proposed personalization system and the known ones are discussed. Also some of the strong points of the system, such as: Connectivity, expandability, interaction abilities between different constituent components are presented.

Chapter 3 Architecture of a XSLT Based product Information Customisation System

3.1 Introduction

This chapter describes the conceptual and physical architectures of a system that matches customer characteristics to product information in order to present a personalized view of the product to the customer. Similar systems are used in applications such as customer product matching and product recommendations e-commerce applications.

The architecture of the proposed system is based on the following concepts:

- **Data Mining.** Is used in order to produce a set of rules which is to be used for the personalization process. The data mining engine is an integral part of the proposed system. It accepts as parameters the user's profile and the product schema, which both of them are represented by the use of Meta data descriptor files and also monitors the interaction between the user and the system, providing a kind of statistical analysis and user interaction pattern recovery. The outcome of this component is a set of XSLT rules which is realized as a Meta data file. This Meta data descriptor file is used to describe the initial relationships between the elements of the product schema and the elements of the customer schema.
- **Customer Profile and Product Description combination** in order to facilitate the Personalization process. This combination is realized by the use of a set of XSLT rules which create associations between customer and product attributes (these attributes are modeled as XML elements)
- **Metadata usage.** In current approach the XML standard is being employed in order to describe the customer profile-s and to represent the structure of products in a web site. XSLT usage for mapping specific user attributes (realized as xml elements) by using rules (realized by the usage of xml tags) to specific product attributes (represented as xml elements) or range of products. XSL transformation rules that encode personalization principles to deliver precise, tailor-made product information to the customers. Product/service information can be described in a neutral and platform independent format (XML) and delivered to the customer over multiple communication media and technologies. Ideally the customer can receive customized product/service information directly to the device of his choice, e.g. a Web browser, mobile phone, digital TV etc. By avoiding the costs of supporting hard coded customization rules and multiple data formats, mass customization can become financially viable.

The structure of this chapter is as follows: It begins with a description of the system architecture at a conceptual level and continues with a brief description of the concepts on which the system is based. Discuss some similarities and differences between the proposed personalization system and the known ones and represents some of the strong points of the system, such as: Connectivity, expandability, interaction abilities between different constituent components. It continues by describing the Conceptual Architecture of the proposed system. It describes the overall architecture of the proposed system. It describes the interaction between the customer and the functional components of the system and also the way that the information is passed from one part of the system to the other part of the system. It describes the different programming components which are working together in order to achieve personalization. Next, the representation of customer profiles and product/service models as XML schemas is covered. The requirements for the specification and modeling of product and customer information are represented. XML file structure is employed in order to model the customer information, his profile, also to represent product information. Some of the commonly used fields for customer profiles are represented. Also some examples with XSLT rules are given. These rules are produced by a Web Mining Tool which is a part of our proposed system. Some components of the system perform complex tasks, such as creating dynamically relationships and extraction rules for a specific customer and a product or range of products. The inner working of this component is discussed. Also in this section there is a different approach discussed Such as Ramakrishnan's approach [Ramakrishnan, 2000]. Then a schematic representation is used in order to describe the workflow steps for matching customer profiles to product information and illustrates the approach with the use of an example of a product personalisation application for the user of an e-commerce web site.

3.2 Conceptual Architecture of the proposed system

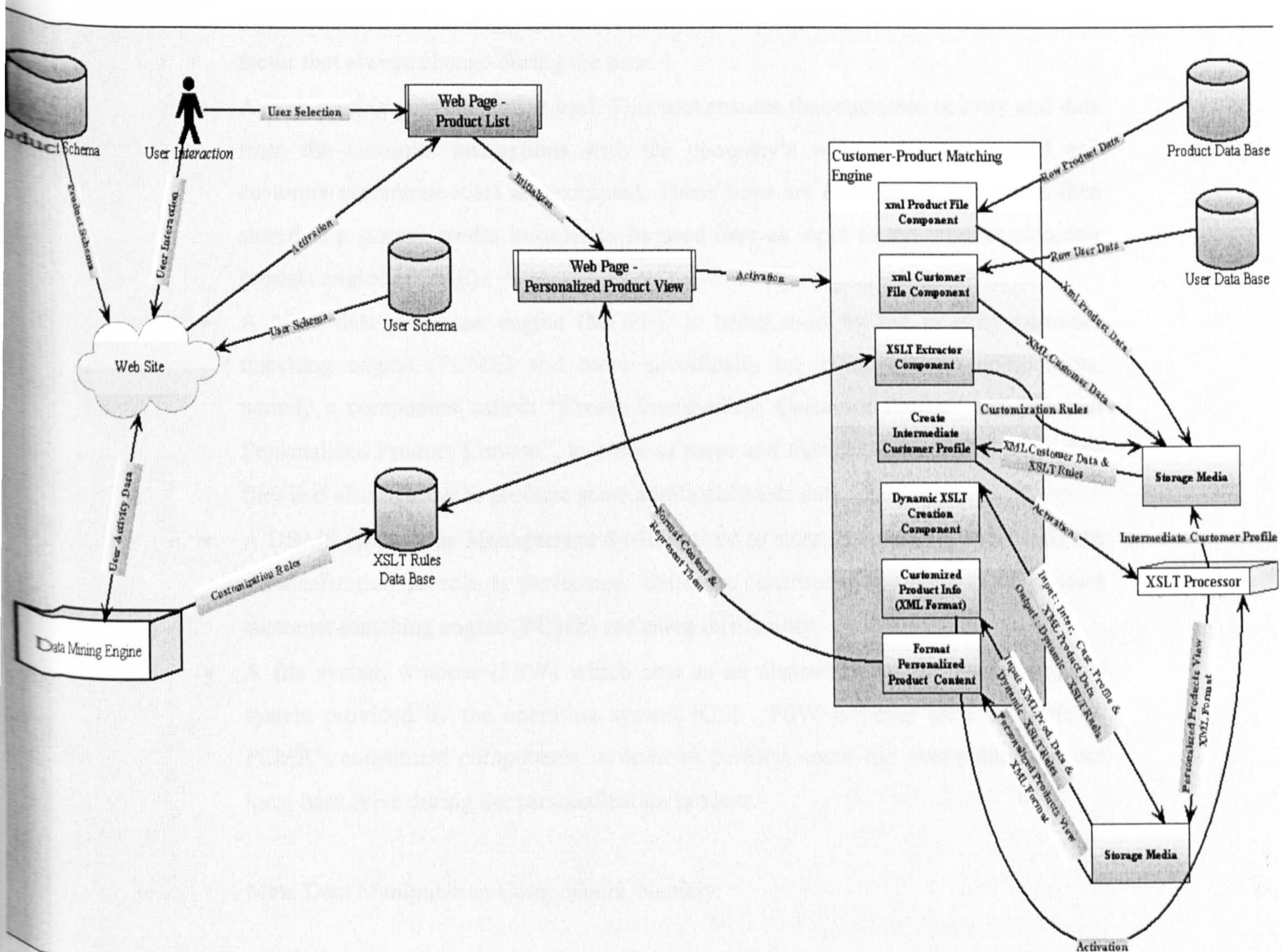


Figure 1: Conceptual Architecture of the proposed system.

The diagram of figure 1 describes the conceptual architecture of the proposed product customer matching engine (PCME) and its context.

The assumption made is that the product-customer matching engine (PCME) is part of an operational customer support environment that additionally incorporates other components such as:

- A web site used for interaction with the customer in his/her product information retrieval/selection and customization activities.
- A product management system (PDM) that maintains both product and customer Meta data. PDM ensures that the product specifications remain up to date (product schema) and that product descriptions (XML instances) are consistent with that at all

times. Also consistency is checked against the customer profile in order to be verified if it is up to date. And especially about customer data which might play a role in the personalization rules. E.g. income, marital status, number of children, which are factors that always change during the time.

- A Web mining personalization tool. This tool ensures that customer activity and data from the customer interactions with the company's web site are analyzed and customer preference rules are extracted. These rules are encoded in XSLT and then stored in a storage media in order to be used later as input to the product customer product engine (PCME).
- A Meta data processor engine (MDPE). Is being used by the product customer matching engine (PCME) and more specifically by its constituent components, namely a component called: "Create Intermediate Customer Profile" and "Format Personalized Product Content", in order to parse and execute some XSLT Meta data files and also in order to produce some additional Meta data files.
- A DBMS (Data Base Management System) used to store intermediate data while the personalization process is performed. Different constituent components of product customer matching engine (PCME) are using this facility.
- A file system wrapper (FSW) which acts as an abstraction of the underlying file system provided by the operating system (OS) . FSW is being used by some of PCME's constituent components in order to perform some file manipulation in the local hard drive during the personalization process.

Meta Data Manipulation Components. Namely:

- Xml Product File Component (XPFC). Extracts product information from the DBMS. The product data are stored in the DBMS as row data in order to reduce the storage space which is required for saving the product information. Then DOM (Document Object Model) [W3C 2004] is used in order to construct the meta-data data structure, which is a tree representation.
- XML Customer File Component (XCFC). Extracts customer information from the DBMS. The customer data are stored in the DBMS as row data in order to reduce the storage space which is required for saving the customer information. Then DOM (Document Object Model) is employed in order to construct the Meta data file, which describes the customer. This Meta data file is an instance of a schema.

- **XSLT Extractor Component (XEC).** Retrieves the XSLT Meta data file which describes the association between a specific customer and a set of products from the DBMS and stores this by using the functionality of FSW in a local storage media, where the internet browser exists and the user interacts with the web site.
- **Create Intermediate Customer Profile (CICP).** An intermediate customer profile is created by this component. Is realized by the use of a Meta data file. It has a tree structure and describes the favorite elements of a product or range of products for a specific customer. Figure 4 shows the structure of the intermediate customer profile. Meta Data Processor Engine (MDPE) and File System Wrapper (FSW) are used by CICP component in order to complete this task.
- **Dynamic XSLT Creation Component (DXCC).** Produces yet another Meta data file, which is an XSLT file which facilitates the extraction of specific favorite elements from a product – range of products for a specific customer.
- **Customized Product Info (CPI).** The Meta Data file produced by Dynamic XSLT Creation Component is being used alongside the product Meta data file in order the component to produce yet another Meta data file describing the personalized product-s view. The Meta Data Processor Engine (MDPE) and File System Wrapper (FSW) are also used by this component.
- **Format Personalized Product Content (FPPC).** Converts the personalized Meta data product file to a readable format. Some html tags are added in order the personalized view to be displayed in a web browser.

All the above Meta Data manipulation components are dealing with Meta data file manipulation. These Meta data files are being produced during the personalization process. Some of them are temporarily created and some others are permanently created and saved in the local storage media. Most of the interaction between the components is done by means of Meta data file exchange. This facilitates the expansion and enhancement of the functionality of the proposed system without many changes.

3.3 Representation of Customer Profiles and Product/Service Models as XML schemas

The requirements for the specification and modeling of product and customer information must amongst others include the ability to cope with:

- Information about customers’ needs, e.g. how important a particular aspect of a product or service is to the customer.
- Issues such as the specification of the “best” possible combination of product features to meet customer’s needs.
- The development of a conceptual product/service model that is understandable to customers, since communication between the organization and its customers is essential.

The product model represents the characteristics of products. Examples of the concepts modeled in the products model include price, availability of a feature e.g. piece of information or mechanical part, or functionality, flexibility, standards, delivery time/place/method, etc.

According to our approach, each product p offered by the organization is of a particular type P . A product p is described by a set of features $f_1, f_2, f_3, \dots, f_n$. Each feature f_k can be *complex* or *simple*. Complex features are further decomposed into other sub-features e.g. f_{k1}, f_{k2}, \dots , which also can be complex or simple. In our approach, product types are implemented as XML Schemas. Individual products are described as XML documents, which are instances of XML schemas. Product features are implemented as elements in an XML schema.

The car product schema (only a part of the entire schema) shown in table 1 below consists of a number of elements that contain information about cars such as body style, boot size, top speed etc.

```
<?xml version="1.0" ?>
- <Schema name="Product_Car.xsd" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType dt:type="string" name="Product_Name" content="textOnly" model="closed" />
  <ElementType dt:type="string" name="bodystyle" content="textOnly" model="closed" />
  <ElementType dt:type="int" name="doors" content="textOnly" model="closed" />
  <ElementType dt:type="float" name="bootsize" content="textOnly" model="closed" />
  <ElementType dt:type="string" name="rated_output" content="textOnly" model="closed" />
  <ElementType dt:type="float" name="acceleration" content="textOnly" model="closed" />
  <ElementType dt:type="int" name="top_speed" content="textOnly" model="closed" />
  <ElementType dt:type="string" name="engine_type" content="textOnly" model="closed" />
  <ElementType dt:type="int" name="engine_size" content="textOnly" model="closed" />
```

<ElementType dt:type="float" name="city" content="textOnly" model="closed" /> <ElementType dt:type="float" name="extra_urban" content="textOnly" model="closed" /> <ElementType dt:type="float" name="combined" content="textOnly" model="closed" /> <ElementType dt:type="int" name="seatbelts" content="textOnly" model="closed" /> <ElementType dt:type="int" name="airbags" content="textOnly" model="closed" /> - <ElementType dt:type="int" name="Esp" content="textOnly" model="closed">
--

Table 1 Specification of a car product in XML

Customer Profile

Depending on the personalization methods used, there are different requirements for the contents and the representation of the profile. Profiles of individual customers are also implemented as XML documents that are instances of XML Schema profiles. Schema profiles describe the profiles of types of customers according to the customer segmentation used by a particular organization. Examples of the concepts represented in a customer profile model include:

- Consumer Income.
- Social habits of the consumer
- Generally, factors that pertain to consumer behavior and reflect customer satisfaction and loyalty.

[Schubert, 2004] classifies the user profiles in two categories:

1. Explicit Profiles.

- 1.1 Identification Profiles. The content of this type of profile can include the following customer information: user name, contact information, personal browser settings, address, payment information, IP-Address etc.
- 1.2 Preference Profile. Includes Self-revealed preferences (product Meta data).
- 1.3 Socio-economic Profile. Self-categorization in predefined classes (age, gender, hobbies etc).
- 1.4 Ratings. This type of user profile has three types of ratings:
 - 1.4.1 Of Products.
 - 1.4.2 Of Reviews.
 - 1.4.3 Of pages.
 And the scale which is used is in the form: I like it or not for me etc.
- 1.5 Relationships. It includes relationships to other users/customers.
- 1.6 Reviews-Opinions. It includes images, plain text, videos and other material.

2. Implicit Profiles.

- 2.1 Transaction Profile.** It might include: Transaction Log, product purchases linked to product meta data (purchases, inquiries, payment)
- 2.2 Interaction Profile.** Its content comprises of click stream data (pages which might have viewed are linked to product meta data [preference categories]).
- 2.3 External Data.** Include information procured from other sources. (e.g. credit rating, events)

Table 2 shows part of a customer profile schema.

<pre> <?xml version="1.0" ?> - <Schema name="CustomerProfile.xsd" xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes"> <ElementType dt:type="string" name="Name" content="textOnly" model="closed" /> <ElementType dt:type="string" name="Address" content="textOnly" model="closed" /> <ElementType dt:type="int" name="Age" content="textOnly" model="closed" /> <ElementType dt:type="string" name="Marital_Status" content="textOnly" model="closed" /> <ElementType dt:type="int" name="NChildren" content="textOnly" model="closed" /> <ElementType dt:type="string" name="Employed" content="textOnly" model="closed" /> <ElementType dt:type="string" name="Employers_Name" content="textOnly" model="closed" /> </pre>

Table 2 Profile of Customers using XML schema

Product Schema transformations

This thesis argues that product and customer information captured in the form of XML models can be utilized for the automatic generation of customized product information. Product model transformation is achieved through the use of rules that:

- describe the degree of customer satisfaction with a particular product feature
- Show which features of a product/service are of interest to the particular customer and to what degree.
- describe the product features that contribute towards customers needs satisfaction, so that products can be customized by providing and/or improving such features
- Present product information to the customer in a way that meets his needs and interests, i.e. presenting only relevant information at a level that is suitable for the particular customer.

A web mining tool which is an integral part of the proposed architecture is contributing in the creation of these associative rules which are encoded by using the XSLT standard. Our approach utilizes XSLT, [W3C 1999] which is a language for transforming XML documents into other XML documents. A transformation expressed in XSLT describes rules for

transforming a source tree into a result tree. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source tree. A template is instantiated to create part of the result tree. The result tree is separate from the source tree. The structure of the result tree can be completely different from the structure of the source tree. In constructing the result tree, elements from the source tree can be filtered and reordered and arbitrary structure can be added.

The thesis' approach is to apply XSL transformation rules to a product specification in order to customize it to match interests of a particular customer. An example XSLT transformation is to remove from the result tree all elements that are not of high interest to a customer according to his profile.

Referring to the car product schema (Table 1) and the Customer profile schema (Table 2) the XSLT rules shown in Table 3 selects relevant product features for different customer types. The rules in Table 3 state that customers who are married and over the age of 45 consider information about a car's safety features and fuel consumption to be 'interesting'.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/Customer_Profile">
    <xsl:if test="/Customer_Profile/Age[.>='45']">
      <xsl:if
test="/Customer_Profile/Family_Status/Marital_Status[.='married']">
        <xsl:element name="profile">
          <xsl:element name="safetyfeatures">Interesting</xsl:element>
          <xsl:element
name="fuel_consumption">Interesting</xsl:element>
        </xsl:element>
      </xsl:if>
    </xsl:if>
  </xsl:template>
</xsl:stylesheet>
```

Table 3: Product Customization Rules

Rules such as the one shown in Table 3 can apply to product specifications in order to customize them according to the profiles of individual customers or types of customers. Applied for example to the description of a specific car model (shown as an XML instance in Table 4), the rule of Table 3 will filter out all car information that does not relate to safety features or to fuel consumption.

```
<Product_Car>
  <Technical_Data>
    <Product_Name>XYZ</Product_Name>
    <bodydescription>
      <bodystyle>Saloon</bodystyle>
      <doors>4</doors>
```



```

        <bootcapacity>500 litres</bootcapacity>
    </bodydescription>
    <enginespecification>
        <rated_output>125/4.200</rated_output>
        <acceleration>9.2</acceleration>
        <top_speed>227kmph</top_speed>
        <engine_type>petrol</engine_type>
        <engine_size>1800cc</engine_size>
        <fuel_consumption>
            <city>9.6</city>
            <extra_urban>5.0</extra_urban>
            <combined>6.7</combined>
        </fuel_consumption>
    </enginespecification>
    <safetyfeatures>
        <seatbelts>5</seatbelts>
        <airbags>4</airbags>
        <Esp>yes</Esp>
        <Asr>yes</Asr>
        <Bas>yes</Bas>
        <side_bags>yes</side_bags>
        <window_bags>yes</window_bags>
        <belt_tensioners>yes</belt_tensioners>
    </security>
</Technical_Data>
</Product_Car>

```

Table 4 XML specification of an individual car model

XSLT rules such as those shown in Table 3 present several benefits, namely:

- The same rule can apply to profiles of individual customers, or types of customers, and to individual products or product categories. There is no need to hardcode rules for individual customers or products.
- The rules make no assumptions about presentation issues (user interface). Once filtered by the customization rules, relevant product information can be appropriately formatted and channeled to the customer over different media (i.e. as email, HTML etc).
- New products can be introduced and product information can be varied without needing to rewrite the customization rules. For example, safety features may vary widely between different car makes. As long as the standardized use of high level naming of elements is obeyed (element 'safety features'), the actual content and organization of sub elements can vary without the need to modify the customization rules.

3.4 Algorithm for creating XSLT Product Feature Selector

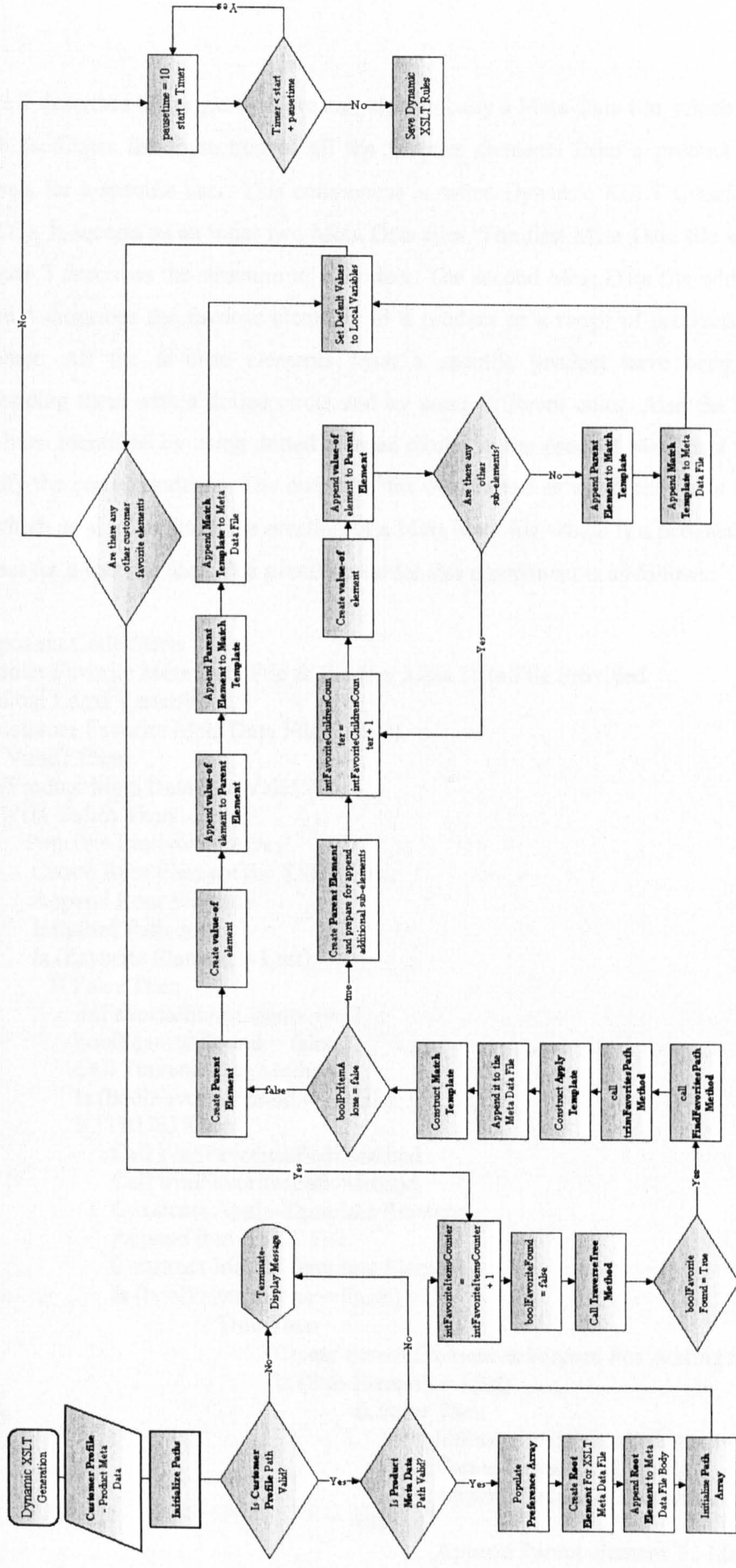


Figure 2 Flowchart describing the algorithm used in order to produce dynamically a Meta data file (using the XSLT standard) which facilitates in the action of all the relevant and favorite product elements of a specific customer.

Figure 2 describes the process of creating dynamically a Meta Data file, which is a XSLT file which facilitates the extraction of all the favorite elements from a product or a range of products for a specific user. This component is called Dynamic XSLT Creation Component (DXCC). It accepts as an input two Meta Data files. The first Meta Data file which is shown in figure 3 describes the structure of a product. The second Meta Data file which is shown in Figure 4 describes the favorite elements of a product or a range of products for a specific customer. All the favorite elements from a specific product have been identified by surrounding them with a dotted circle and by using different color. Also the same elements have been identified by using dotted colored circles in the product Meta data file in order to identify the correspondence. The output of the component as we already said is a Meta Data file which used to facilitate the creation of a Meta Data file which is a personalized view of a product for a specific user. The pseudo code for this component is as follows:

```

Component Code Starts
Customer Favorite Meta Data File & Product Meta Data File Provided
Set Initial Local Variables
Is (Customer Favorite Meta Data File Valid?)
If (Is Valid) Then
    Is (Product Meta Data File Valid?)
    If (Is Valid) Then
        Populate Preference Array
        Create Root Element for XSLT File
        Append Root Element
        Initialize Path Array
        Is (Favorite Element = Last)
        If False Then
            intFavoriteItemsCountr += 1
            boolFavoriteFound = false
            Call TraverseTree Method
            Is (boolFavoriteFound = TRUE)
            If (TRUE) Then
                Call FindFavoritesPath Method
                Call trimFavoritesPath Method
                Construct Apply-Template Element
                Append it to XSLT File
                Construct Match-Template Element
                Is (boolPrItemAlone = False)
                If True Then
                    Create Parent Element & Prepare For Adding Sub Elements
                    Is (Sub-Element = Last)
                    If False Then
                        intFavoritechildrenCounter += 1
                        Create Value-Of Element
                        Append Value-Of Element to ParentElement
                    Else
                        Append Parent element To Match Template

```

```

Append Match Template To XSLT Body
Set Default Values To Local Variables
End IF
Else
Create Parent Element
Create Value Of Element
Appent Value Of Element To parent Element
Appent Parent Element To Match Template
End If
Else
Set PauseTime = 10
Start = Timer
Is Timer < Start + PauseTime
If True Test Again
Else
Save Dynamic XSLT Rules
Terminate Execution
End If
End If
Else
Terminate Execution– Display Message
Else
Terminate Execution – Display Message

```

What this code does is:

1. first parses the favorite elements meta data file
2. extracts all the elements which are considered favorites for a specific customer
3. then goes and finds all the corresponding elements in the product meta data file
4. constructs a meta data file which is again represented as a tree structure and describes the position of the favorite elements inside the product meta data file and also the way which has to be used in order these elements to be extracted.

This component's code applies the ideas taken by a different personalization approach described by Ramakrishnan, [Ramakrishnan, 2000]. In Ramakrishnan's approach a structure of a web site is abstracted by a program. The same approach is applied here but in a different way and in different fashion. The abstraction is scaled down and is used to describe products instead of web sites. The difference here is that we don't need a special program which traverses the web site in order to identify its structure. A proposed standard [W3C 2004] is being used in order to manipulate the content of the product which is represented as a tree structure. Also in Ramakrishnan's approach the personalization is achieved by passing to the program which abstracts the web site some variables, which are describing the "virtual route" that the user takes in the web site, and then reducing the size of the program. In this approach the user doesn't need to perform some specific selection on a product, or to specify some elements of a product. The favorite elements are described by a Meta data descriptor file. This file looks as a scaled down version of the entire product and is shown in Figure 4. The

structure of the entire product is shown in Figure 3. The idea now is to create a process which traverses the Meta data file descriptor and extracts all the elements which are favorites and then create another Meta data file which actually describes the way of pinpointing these elements and the way of extracting them. There are different colored circles in both the Meta data descriptor file and in product Meta data file which show the correspondence between the elements in these two Meta data files. In most personalization scenarios we experience shrinkage of the original product Meta data file. The personalization process results to a Meta data file, a dynamically created set of XSLT rules which extract all the desired user elements alongside with their child elements. E.g. when e1 is to be extracted from the product Meta data file is not going to be extracted alone. Also its sub-elements namely e.1.1, e.1.1.1, e.1.1.2, e.1.2, e.1.2.1, e.1.2.2 are also to be extracted.

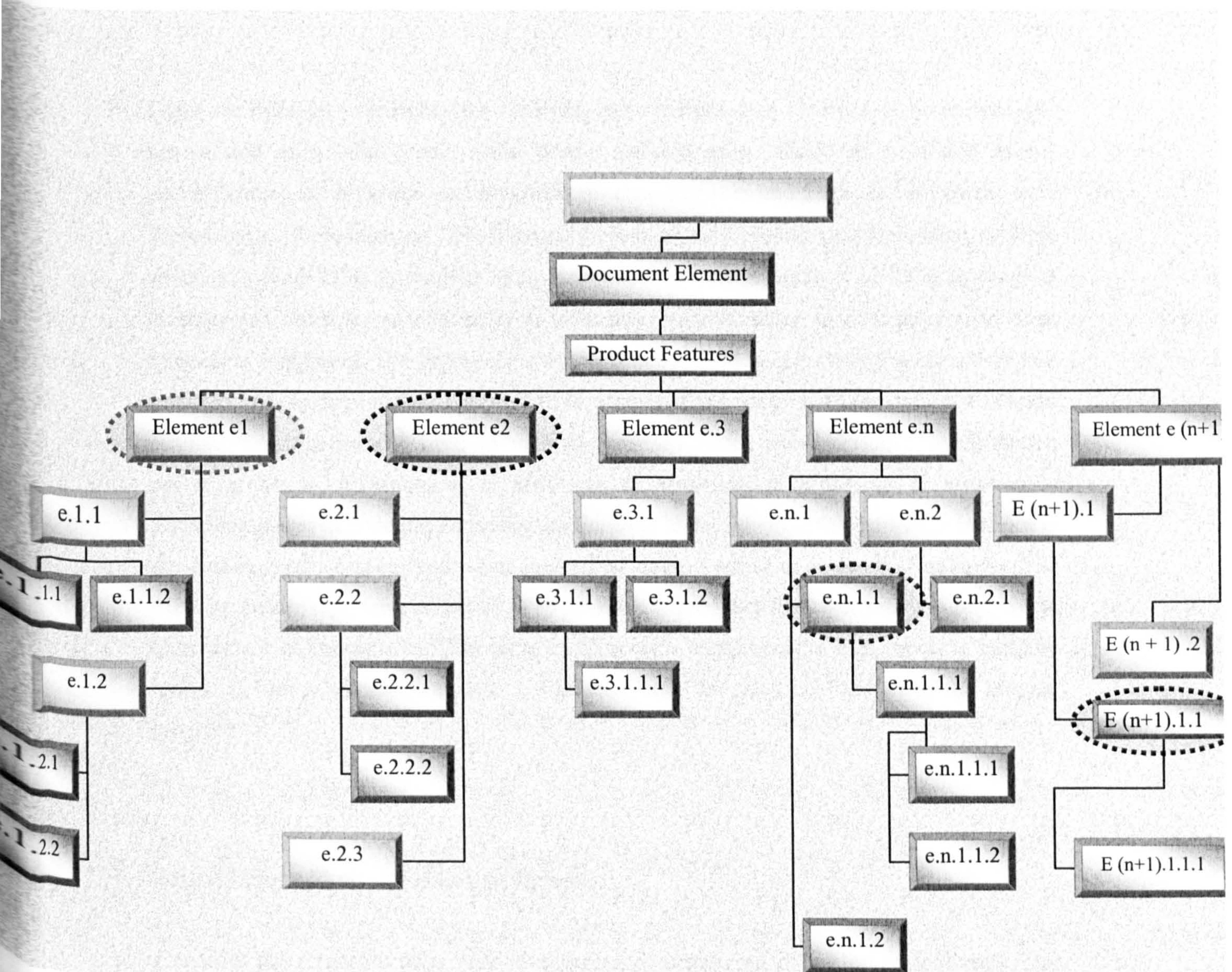


Figure 3 Product Meta Data File. This is a tree representation of the product. The circles in the diagram indicate the favorite elements for a specific customer.

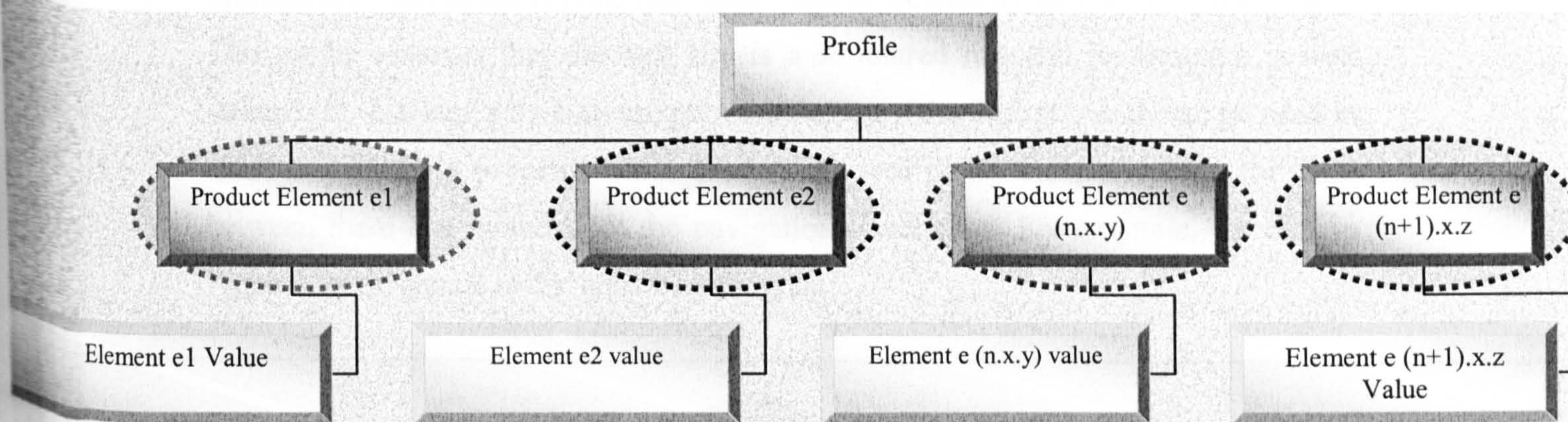


Figure 4 Intermediate profile. Is a meta data file which describes which elements of the product attract interest of a specific customer.

There are different approaches for achieving personalization in a web site. Some are well known and established ones while others are emerging. Most of them are hybrid personalization methods and combine different characteristics and methods used by the well known ones. [Ramakrishnan, 2000] describes one of these hybrid personalization systems which is called PIPE (short for “Personalization is Part Evaluation”). PIPE employs a technique in order to personalize Web resources, without taking in account the previous interaction sequences. The input to a partial evaluator (which is a type of a compiler) is a program and its arguments. The output is a customized version of the program. Partial evaluation is traditionally used to make a program faster and to remove interpretation operating cost. Is a technique which facilitates the simplification of program presentation by discarding pieces of code implementing the business logic (which in this particular instance is the customization or personalization logic) that doesn’t apply to a specific customer. The PIPE approach represents a set of Web sites as a program that abstracts the underlying organization of information. The program then with respect to the user input, is partially evaluated, and a tailored Web site is produced from the short cut version of the original program.

Abstracting a Web site into a program

Consider the structure of a Web site, which is organized in a hierarchical way. The web site is abstracted as a tree representation. In Figure 5, the nodes (rectangles) represent individual Web pages for the site and the links represent some tagging

mechanism, such as the HTML (an <href> that associates hyperlinks in a Web page). This model assumes that the web site is a structured one and its structure is well defined. In this case a parsing program employing a depth first search can be used in order to produce a program which models the web pages themselves and the links between them and includes all the navigational scenarios. For example, the data in Figure x corresponds to the following program:

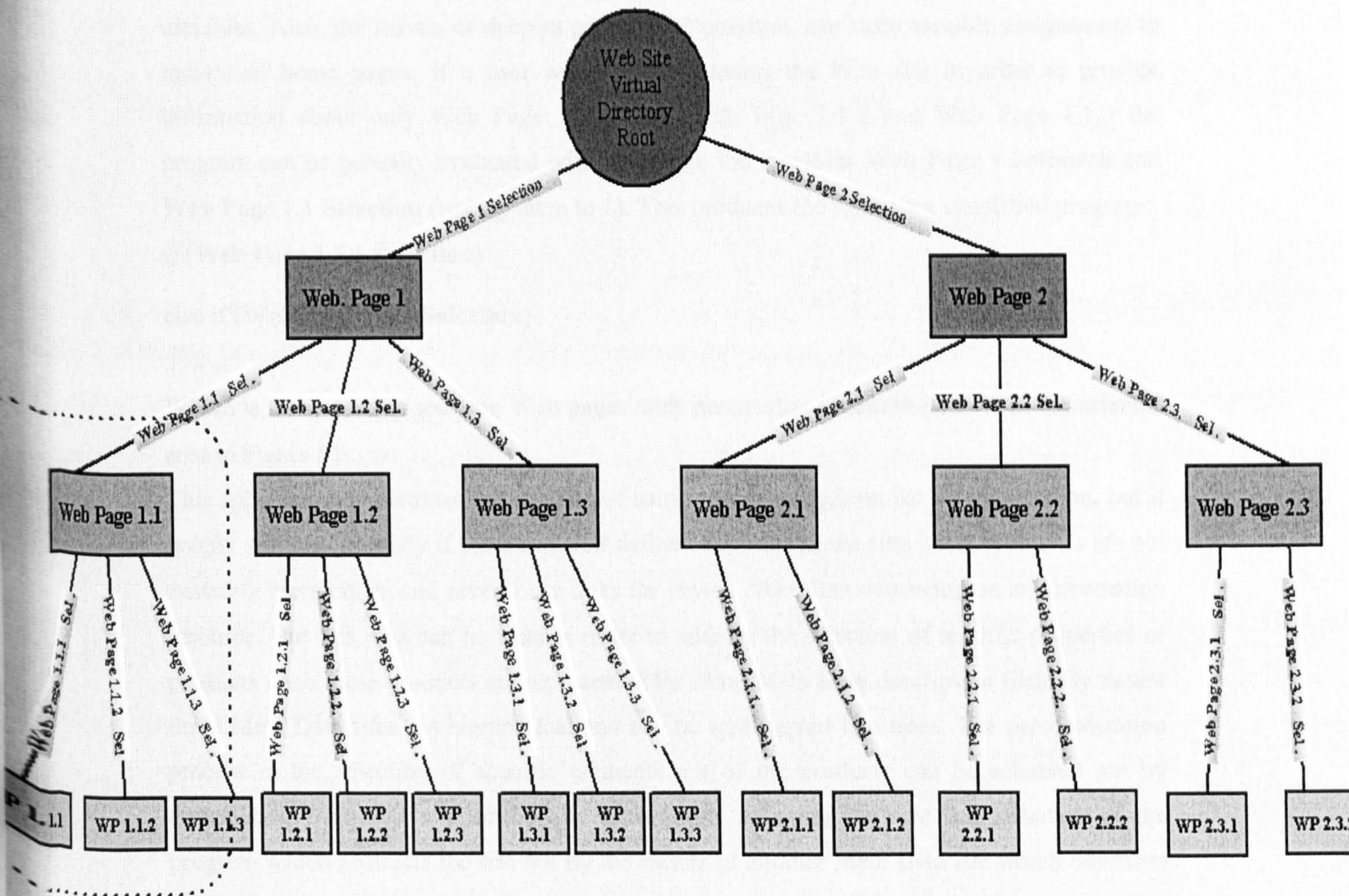


Figure 5. First three levels of a hierarchically organized Web site. Lower levels could include detailed information. The labels on edges represent selections made by a user.

```

if (Web Page 1 Selection)
    if (Web Page 1.1 Selection)
        if (Web Page 1.1.1 Selection)
            ...
        else if (Web Page 1.1.3 Selection)
            ...
    
```



```

        else if (Web Page 1.2 Selection)
            ...
...
else if (Web Page 2 Selection)
    if (Web Page 2.1 Selection)
        ...

```

The link labels are represented as program variables. The program models the links at individual nodes as mutual exclusives by employing else if statements. Even though this model depicts only the organization of the Web site, the information stored at each of the inner nodes can be modelled by associating complicated data structures with the program variables. Also, the *leaves*, or deepest parts of the program, can store variable assignments to individual home pages. If a user wants to personalize the Web site in order to provide information about only Web Page 1.1.1. And Web Page 1.1.2 and Web Page 1.1.3 the program can be partially evaluated with respect to the variables **Web Page 1 Selection** and **Web Page 1.1 Selection** (setting them to 1). This produces the following simplified program:

```

if (Web Page 1.1.1 Selection)
    ...
else if (Web Page 1.1.3 Selection)
    ...

```

Which is to be used to recreate Web pages with personalized content (shown in the selected area in Figure 5).

This approach demonstrates the concept of using partial evaluation for personalization, but it is only working properly if there is a well defined structure in the site. Most Web sites are not austere hierarchical and several use links for reason other than narrowing on an information resource. But this idea can be used in order to address the selection of specific properties of products once these products are represented by using Meta Data description files. By nature these Meta Data files are hierarchical and can be represented like trees. The personalization process in the selection of specific elements out of the products can be achieved not by anticipating each user's selection and accordingly reducing the size and structure of the program which abstracts the site but by the means of another Meta Data file which describes the preferences of the specific user.

3.5 Workflow for Customer Product Matching System

This section describes the workflow steps for matching customer profiles to product information and illustrates the approach with the use of an example of a product personalisation application for the user of an e-commerce web site.

Workflow steps

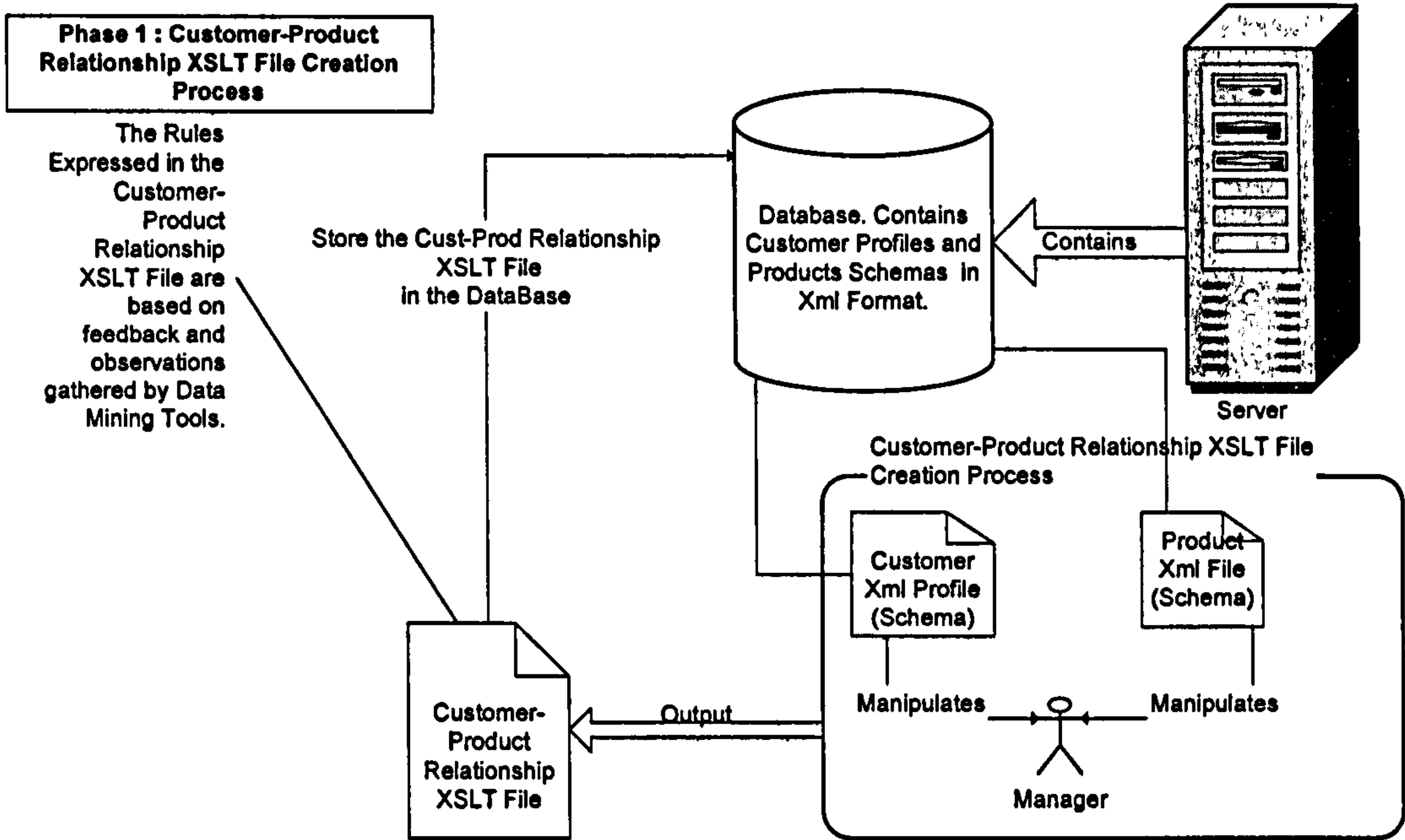


Figure 6 STEP 1: Create Customer-Product Relationship XSLT File is created.

Although the terms ‘document’ and ‘document schema’ are typically used to refer to XML data, in fact, neither instances nor schemas need to exist as documents per se. XML data may exist as streams of bytes sent between applications, or as fields in database records etc. A prototype system for the mass customization approach described here has been implemented as a proof of concept. XML documents for product descriptions and customer profiles have been implemented as records in a relational database using a commercial database management system (DBMS). The DBMS supports the creation of XML documents ‘on the fly’ using data stored in the tables. This approach is also recommended for a commercial realization of this system as the typical large number of customers (profiles) and products would make a file based approach (i.e. where each individual product or profile is stored in a separate XML file) impractical. An XSLT processor is an application that applies XSL transformations to an XML file returning another file (in XML or other format) as output. We have experimented with different XSL processors.

The XSLT output is typically another XML document. However other alternative formats such as HTML, text etc that can be generated by the XSLT processor. An HTML output for example is the preferred format when it has to be sent to a Web browser so that it can be viewed by the user.

The format of the files is as follows:

```
<?xml version="1.0"?>
<Schema name="Customer_Schema.xsd"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">

  <ElementType name="Age" content="textOnly" dt:type="float" model="closed"/>
  <ElementType name="Family_Status" content="textOnly" dt:type="string"
    model="closed"/>
  <ElementType name="ChildrenNumber" content="textOnly" dt:type="integer"
    model="closed"/>
  <ElementType name="Job_Title" content="textOnly" dt:type="string"
    model="closed"/>
  <ElementType name="Financial_Ability" content="textOnly" dt:type="string"
    model="closed"/>
  <ElementType name="Customer_Profile" model="closed" content="eltOnly"
    order="seq">
    <element type="Age" minOccurs="1" maxOccurs="1"/>
    <element type="Family_Status" minOccurs="1" maxOccurs="1"/>
    <element type="ChildrenNumber" minOccurs="1" maxOccurs="1"/>
    <element type="Job_Title" minOccurs="1" maxOccurs="1"/>
    <element type="Financial_Ability" minOccurs="1" maxOccurs="1"/>
  </ElementType>
</Schema>
```

Table 5: Customer XML Profile (Schema)

```
<?xml version="1.0"?>
<Schema name="Car_Schema.xsd"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">

  <ElementType name="bodystyle" content="textOnly" dt:type="string"
    model="closed"/>
  <ElementType name="doors" content="textOnly" dt:type="integer"
    model="closed"/>
  <ElementType name="bootcapabilities" content="textOnly" dt:type="string"
    model="closed"/>
  <ElementType name="bodydescription" model="closed" content="eltOnly"
    order="seq">
    <element type="bodystyle" minOccurs="1" maxOccurs="1"/>
    <element type="doors" minOccurs="1" maxOccurs="1"/>
    <element type="bootcapabilities" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="enginetype" content="textOnly" dt:type="string"
    model="closed"/>
```



```
<ElementType name="enginesize" content="textOnly" dt:type="string"
model="closed"/>
<ElementType name="enginespec" model="closed" content="eltOnly" order="seq">
    <element type="enginetype" minOccurs="1" maxOccurs="1"/>
    <element type="enginesize" minOccurs="1" maxOccurs="1"/>
</ElementType>
<ElementType name="seatbelts" content="textOnly" dt:type="integer"
model="closed"/>
<ElementType name="airbags" content="textOnly" dt:type="integer"
model="closed"/>
<ElementType name="security" model="closed" content="eltOnly" order="seq">
    <element type="seatbelts" minOccurs="1" maxOccurs="1"/>
    <element type="airbags" minOccurs="1" maxOccurs="1"/>
</ElementType>
    <ElementType name="features" model="closed" content="eltOnly"
order="seq">
        <element type="bodydescription" minOccurs="1" maxOccurs="1"/>
        <element type="enginespecification" minOccurs="1" maxOccurs="1"/>
        <element type="security" minOccurs="1" maxOccurs="1"/>
</ElementType>
<ElementType name="car" model="closed" content="eltOnly" order="seq">
    <element type="features" minOccurs="1" maxOccurs="1"/>
</ElementType>
</Schema>
```

Table 6: Product XML File (Schema)

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/ Customer_Profile ">
        <xsl:if test=" Family_Status [.='Not Married']">
            <xsl:if test="Age[.='30']">
                <xsl:element name="enginespec">Interesting</xsl:element>
                <xsl:element name=" bodystyle">Interesting</xsl:element>
                <xsl:element name="security">Interesting</xsl:element>
            </xsl:if>
            <xsl:if test="Age[.='45']">
                <xsl:element name=" enginespec">Not
                Interesting</xsl:element>
            </xsl:if>
        </xsl:if>
    </xsl:template>
</xsl:stylesheet>
```

Table 7: Customer-Product Relationship XSLT File

Now, when the user wants to enter and do some shopping he has to log in to the web site. This is the second phase of the process. A diagram follows which describes the sequence of events.

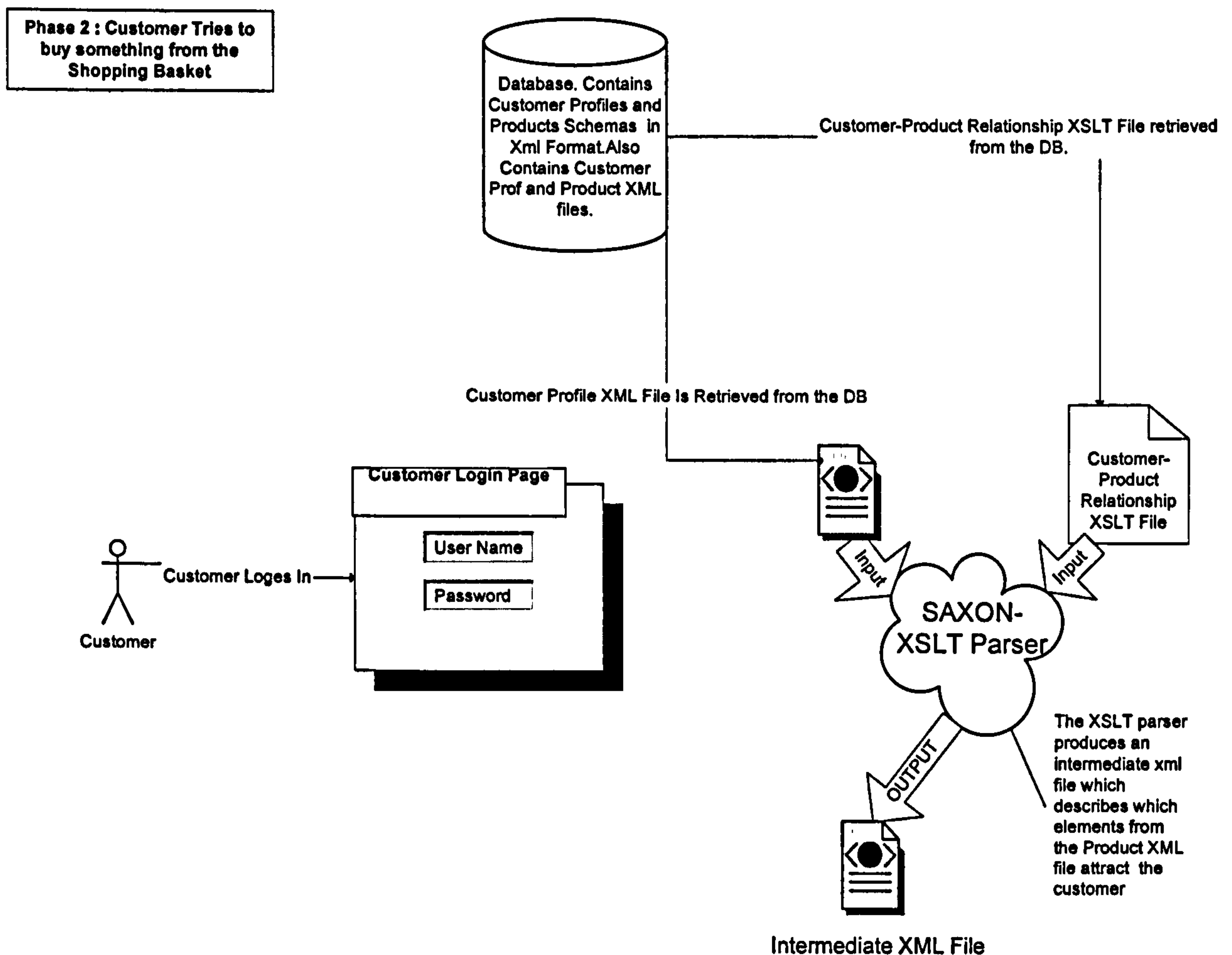


Figure 7 STEP 2: User Logs In. a sequence of events is taking place.

Once the user loges in his id is checked against the existing ids in the database in order to verify whether we have or not an associated Customer-Product Relationship File that describes the types of elements belonging to a specific product category might attract the customer. If this is the case, then the Customer Profile XML File is loaded. An example of this file can be seen bellow:

```

<Customer_Profile>
  <Age>30</Age>
  <Family_Status>Not Married</MaritalStatus>
  < ChildenNumber>0</ ChildenNumber>
  < Job_Title>Programmer</ Job_Title>
  < Financial_Ability>Good</ Financial_Ability>
</Customer_Profile>
  
```


Table 8: Customer Profile XML File.

Also the Customer-Product Relationship XSLT File is loaded. A description of this file can be seen in Table 7.

Once the two files are loaded then we execute a XSLT parser which has as its input the two files. We do this step in order to produce an intermediate file which describes all the elements that the user might find interesting. The output of the XSLT processor can be seen below:

```
<profile>
  < enginespec >interesting</ enginespec >
  < bodystyle >interesting</ bodystyle >
  <security>interesting</security>
</profile>
```

Table 9: Intermediate XML File.

When the user selects a product from the product list then a new sequence of events is taking place. All the activities are described in Figure 9.

The Intermediate file and the Product XML File, whose description can be seen below, in Table 10, are parsed from a “Product Customization Process” which produces again one XSLT file which I call it **Customized XSLT**. This Customized XSLT (which can be seen in Figure 8) finally is parsed once again with the use of a XSLT processor alongside with the Product XML File in order to produce the customized information for the product. One of the characteristic of the entire process is that most of the processes produce Intermediate Files and XSLT files and they don’t extract information directly. This is done because is much more effective and efficient to extract information using a XSLT parser, instead of using some other executables. This is more important when we take into consideration the fact that in Internet applications there is a lot of trafficking and a lot of interaction. The execution of many processes together can be very costly.

```
<car>
  <features>
    <bodydescription>
      <bodystyle>Saloon</bodystyle>
      <doors>4</doors>
      <bootcapabilities>500 litres</bootcapabilities>
    </bodydescription>
    <enginespec>
      <enginetype>petrol</enginetype>
      <enginesize>1800 cc</enginesize>
```

```

        </enginespec>
        <security>
            <seatbelts>4</seatbelts>
            <airbags>4</airbags>
        </security>
    </features>
    <Element1>Sample Data</Element1>
</car>

```

Table 10: Product XML File.

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version="1.0">
    <xsl:template match="/">
        <car>
            <xsl:apply-templates select="/car/features/enginespec" />
            <xsl:apply-templates select="/car/features /bodystyle" />
            <xsl:apply-templates select="/car/features /security" />
        </car>
    </xsl:template>
    <xsl:template match=" enginespec ">
        < enginespec>
            < enginetype>
                <xsl:value-of select=" enginetype" />
            </ enginetype>
            < enginesize >
                <xsl:value-of select=" enginesize " />
            </ enginesize >
        </ enginespec >
    </xsl:template>

    <xsl:template match=" bodystyle ">
        < bodystyle >
            <xsl:value-of select=" bodystyle " />
        </ bodystyle >
    </xsl:template>

    <xsl:template match=" security ">
        < security >
            < seatbelts >
                <xsl:value-of select=" seatbelts " />
            </ seatbelts >
            < airbags >
                <xsl:value-of select=" airbags " />
            </ airbags >
        </ security >
    </xsl:template>
</xsl:stylesheet>

```

Figure 8: Customized XSLT

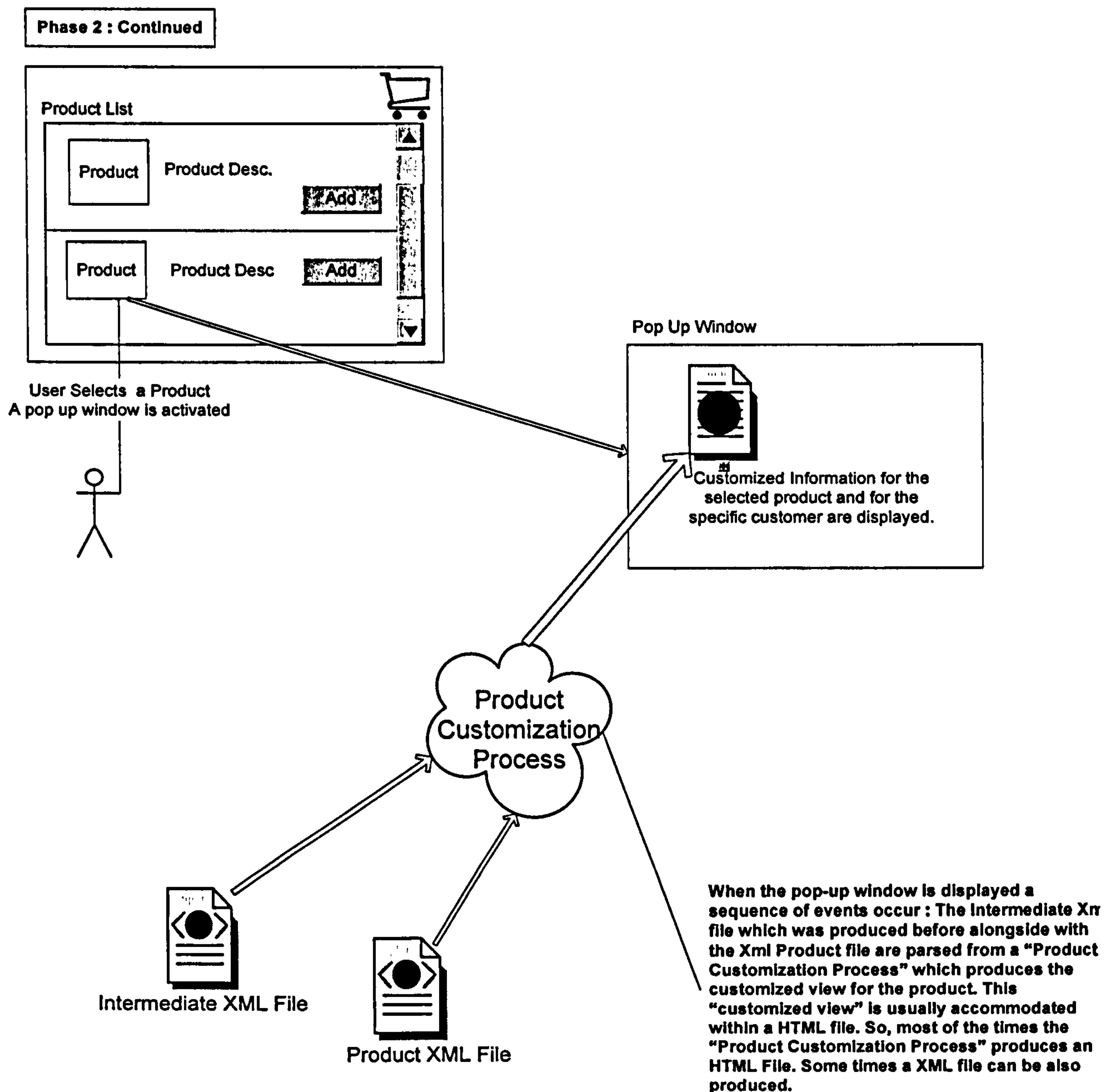


Figure 9: The final sequence of activities in order the user to see the customized information for a product category.

3.6 Conclusion

We have described the system architecture at a conceptual level and then a brief description of the concepts on which the system is based was given. Some similarities and differences between the proposed personalization system and the known ones were discussed and some of the strong points of the system were presented, such as: Connectivity, expandability, interaction abilities between different constituent components. A conceptual architecture of the proposed system was given. The conceptual architecture described the overall architecture of the proposed system. It described the interaction between the customer and the functional components of the system and also the way that the information is passed from one part of the system to the other part of the system. It described the different programming components which are working together in order to achieve personalization. Next, the representation of customer profiles and product/service models as XML schemas was covered. The requirements for the specification and modeling of product and customer information were represented. XML file structure was employed in order to model the customer information, his profile, also to represent product information. Some of the commonly used fields for customer profiles were represented. Also some examples with XSLT rules were given. These rules were produced by a Web Mining Tool which is a part of our proposed system. Some components of the system perform complex tasks, such as creating dynamically relationships and extraction rules for a specific customer and a product or range of products. The inner working of this component was discussed. Also in this section there is a different approach discussed Such as Ramakrishnan's approach [Ramakrishnan, 2000]. Then a schematic representation was used in order to describe the workflow steps for matching customer profiles to product information and illustrates the approach with the use of an example of a product personalisation application for the user of an e-commerce web site. The structure of the Customer and product xml files has been described. The association of the customer profile with the product description through the use of XSLT rules has been described. Also the different transformations that occur through the process of customization have been described. The user- system interaction has also been described. In the next chapter we will represent a more formal view of the proposed system by employing a set of different kind of UML diagrams. Use Case diagrams, class diagrams, sequence and activity diagrams, deployment diagrams are to be used in order to describe in a more detailed way the inner working of the proposed system, its interaction with the users and also the way that this component it might deployed in order to perform better.

Chapter 4 Detailed Design and Implementation of the Personalization System

4.1. Introduction

This chapter describes the design and the implementation of a system that matches customer characteristics to product information in order to present a personalized view of the product to the customer. It begins with an overview of the application software which facilitates the creation of the design models. Lists the different models used in order to describe the overall architecture and design of the proposed demo system. Namely Use Case, Analysis, and Design model are represented. Also the Component view is included. This model shows how the logical components realized by the usage of a programming language and also describes the interrelationship of the different components. The proposed demo system comprises two different set of programs. There is a description of these two different programs alongside with the description of the technologies which have been used in order to realize the proposed system. Then a demo is presented. There are two different users who create accounts in n online shop and then they decide to purchase a product. On purpose the scenario shows that they both buy the same product but the personalization process produces different results for the customers since they are having different customer profiles and different business rules are applied on these profiles in order to produce a personalized feeling in the interaction with the system.

UML (Unified Modeling Language) comes with a graphical environment created by Rational Rose. It facilitates the creation of the different models partially described below.





There are four Views in a Rose Model. [Oestereich, 1999]

1. The Use Case View (Which actually contains the Use Case Model)
2. The Logical View (Which contains the Analysis Model and the Design Model)
3. The component View (Which contains the Component Model)
4. The Deployment View (Which contains the Deployment Model)

Use case View

The Use Case view includes all of the actors, use cases, and Use Case Diagrams in the system. The Use Case view is an implementation-independent look of the system. It focuses on a high level picture of what the system will do, without worrying about the details of how the system will do it.








The Use Case View Includes:


-  **System User** Actors, which are external entities that interact with the system being built. Can be people, or can be other program interacting with the system being built.
-  Use Cases. Are being used to capture functional requirements with a focus on value added to the user. Also they drive the whole development process since most activities such as analysis, design and test are performed starting from the use cases. Design and test can also be planned and coordinated in terms of use cases.
-  Use Case diagrams, which show the actors, the use cases, and the interaction between them. There are typically several Use Case diagrams per system, each showing a subset of the actors and / or use cases.
-  Packages, which are groups of use cases and / or actors. A package is a UML mechanism that helps you to group similar items together. The Use Case Model here is a package, which contains some use cases.

Logical View

The Logical View focuses on how the system will implement the behaviour in the use cases. It provides a detailed picture of the pieces of the system, and describes how the pieces interrelate. The Logical View includes among other things, the Analysis Model and the Design Model.

The Logical View Includes:









-  Design Classes, are the building blocks for a system. Design classes are language specific, while Analysis Classes are more generic. They are directly linked to Analysis Classes (described latter), which are used in the Analysis Model.
-  Class Diagrams. Are being used to view the classes of the system, their attributes and their operations and their relationships to each other. In Analysis Model the Class Diagrams describe the relationship between the classes but don't actually show any of the attributes or member functions.
-  Collaboration Diagrams. These kinds of diagrams show how the focus moves from object to object as the use case is performed and the messages that are sent between the objects. A message sent from one-object triggers the receiving object to take over focus and to carry out one of the responsibilities of its class. Time is not a factor to these kinds of diagrams.
-  Sequence diagrams. Sequence diagrams are Interaction diagrams that are ordered by time.
-  Packages, which are groups of related classes.
-  Boundary Class used to describe an interface in the system. Is being used in the Analysis Model.
-  Entity Class used to describe a storage medium. Is being used in the Analysis Model.

-  Control Class is being used to describe a process and is being used in the Analysis Model.

Component View

The Component View contains information about the code libraries, executable files, runtime libraries, and other components in your model. A *Component* is a physical module of code. In Rose, components and Component Diagrams are being displayed in the Components View. The Component View of the System allows you to see the relationships between the modules of code.

The Component View Includes:

-  Component. The Component icon is used to represent a software module with a well-defined interface. (These icons are being used to represent ActiveX Dll's VB programs)
-  Subprogram Specification. Is being used to represent a subprogram's visible specification.
-  Subprogram body. Is being used to represent the implementation body.
-  Main Program. A main program is the file that contains the root of the program.
-  Package Specification. Is generally a header file, which contains function prototype information for the class.
-  Package Body. Contains the implementation of the functions declared in the Package Specification.
-  ,  Task Specification. And Task Body. These icons are used to represent packages that have independent threads of control. These icons are being used in order to represented ActiveX Exe projects (In VB Programs)

Deployment View

The final view in Rose is the Deployment View. The Deployment View is concerned with the physical deployment of the system.

4.2. System Realisation models

4.2.1 Use Case Model

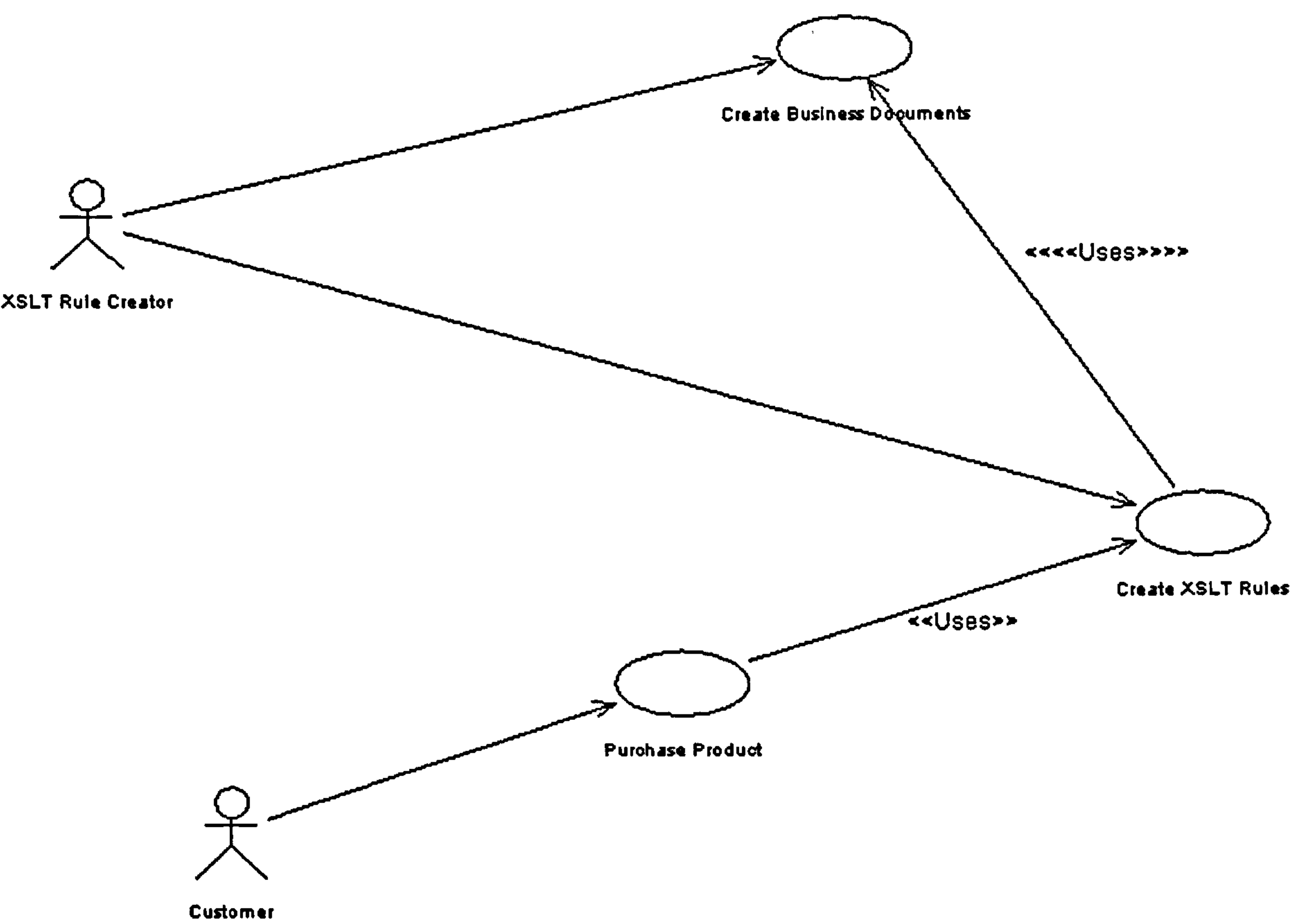
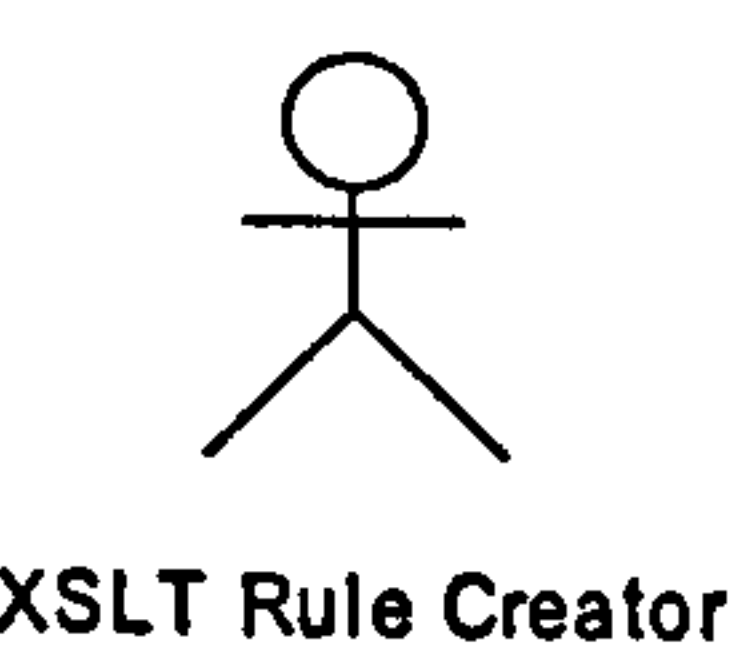
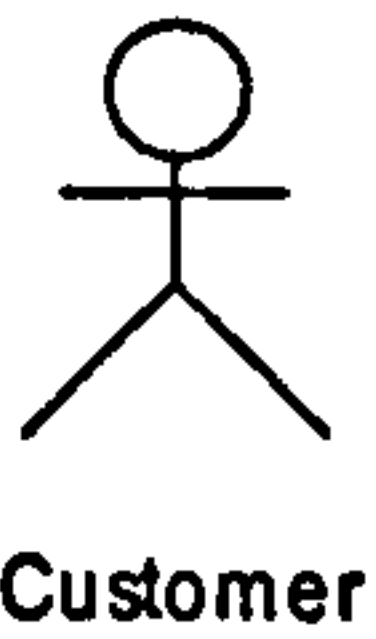


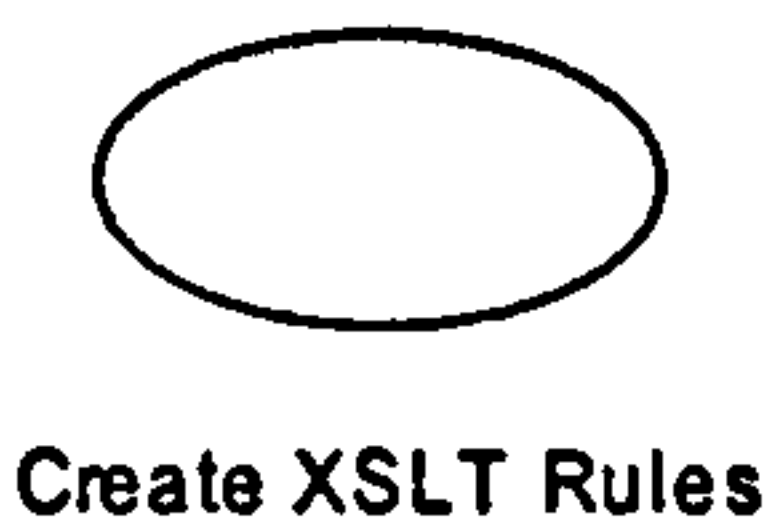
Figure 1: Package Overview



This is the person who actually creates the XSLT Rules which are to be used in the personalization process. The use cases have to be activated in a specific order because the output of one use case is being used as an input to another use case.



This is the actor who purchases products from an online shop. All the XSLT rules created from the XSLT Rules Creator actor are being used in order this actor to see a customized view of the products that he/she wants to purchase. The use cases have to be activated in a specific order because the output of one use case is being used as an input to another use case.



Brief Description

The use case "Create XSLT Rules" is used by the XSLT Rule Creator actor in order to create a set of XSLT rules for a specific customer segment which are going to be used in the personalization process. This set of XSLT Rules is to be saved later in the SQL Server in the table named tblXSLTRules.

Initial Step-By-Step Description

Preconditions

Before the use case is initiated the following actions have to be performed:

- a. The business documents for the Product Category and for the Customer Segment has to be created (in order to describe these documents we are using XML schemas)
- b. The XSLT Rules'Creator user activates the Main Interface.
- c. From the action bar displayed in the top of the window selects the "Create XML Schema" action.

Steps Performed in Create XML Schema Use Case

The XSLT Rules Creator performs the following steps:

1. From the action bar displayed in the top of the window selects the "Customer Schema" action.
2. A dialog box titled "Select Customer XML Schema File" appears. A Customer Schema is selected and then the Command Button Open is selected. The Customer Schema file is loaded and displayed in a tree view control.
3. From the action bar displayed in the top of the window selects the "Product Schema" action.
4. A dialog box titled "Select Product XML Schema File" appears. A Product Schema is selected and then the Command Button Open is selected. The Product Schema file is loaded and displayed in a tree view control.
5. Expands the Tree View controls for the Product and the Customer Schemas.
6. From the action bar displayed in the top of the window selects the "Operations" action.
7. A new Form titled Operations appears.
8. Right Click inside the MDI (Multiple Document Interface). A pop up window appears. Three options are available. If Nested, Choose, If Then Else. Select One of this.
9. Selects some elements from the Customer Schema and drag them into a grid control which is inside the Operations Form. Depending on the previous selection the user can drop one or more items inside the Operations grid control.
10. Selects some elements from the Product Schema and drag them into a grid control which is inside the Operations Form.
11. Select an operation from the left pane located inside the Operations Form. The available operations are: **If, Choose, If Then Else.**
12. Depending on the Operation selected a new form appears. Create the rules and select the Command Button **Append**. The created XSLT rules appear at another window titled XSLT rules.
13. From the action bar displayed in the top of the window selects the "Save XSLT" action. The rules are saved in the SQL server.



Create Business Documents

Brief Description

The use case "Create Business Documents" is used by the XSLT Rule Creator actor in order to create a set of XML Schema files in order to describe the Customer Profile and Product Category. Both files can be created by using some XML Schema Editors (Such as PFE, Or the Microsoft Visual Studio IDE).

Initial Step-By-Step Description

Preconditions

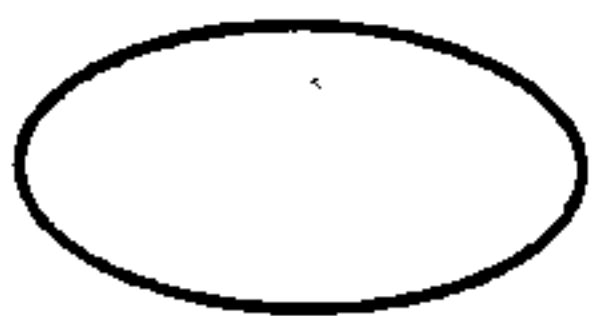
Before the use case is initiated the following actions have to be performed:

None.

Steps Performed in Create XML Schema Use Case

The XSLT Rules Creator performs the following steps:

1. Open any XML Schema Creator (such as PFE, or the Microsoft Visual Studio IDE). Create the Schema and then save them.



Purchase Product

Brief Description

The use case "Purchase Product" is used by the Customer actor in order to purchase a product from an online shop.

Initial Step-By-Step Description

Preconditions

Before the use case is initiated the following actions have to be performed:

- a. The business documents for the Product Category and for the Customer Segment has to be created (in order to describe these documents we are using XML schemas)
- b. The XSLT Rules for the Product Categories and the Customer Segments they should have been created already.

Steps Performed in Create XML Schema Use Case

The Customer actor performs the following steps:

1. If the Customer has a Password then does the following :
 1. 1. Enters the Customer ID
 1. 2. Enters the Customer Password.
 1. 3. Clicks at the Login Command Button. Then is being transferred to the Product List
page.
 1. 4. Selects a product. Then a customized version of this product is being displayed.
2. If the customer doesn't have a password then he/she does the following:
 2. 1. Selects the "Click Here" link button. Then he/she is transferred to the Customer Registration page.
 2. 2. The Customer enters his/hers Customer ID.
 2. 3. The Customer enters his/hers Password.
 2. 4. The Customer enters his/hers Marital Status.
 2. 5. The Customer enters his/hers Sex.
 2. 6. The Customer enters his/hers No Of Dependants.
 2. 7. The Customer enters his/hers Income.
 2. 8. Then Selects the Save Command Button. And is redirected to the product list
page.
 2. 9. Selects a product. Then a customized version of this product is being displayed.

4.2.2 Analysis Model

The Analysis Model is represented by an Analysis System denoting the top level package of the system. Other analysis packages is a way of organizing the analysis model into more manageable pieces that represent abstractions of sub systems. I will be using some subsystems for packaging the Use Case Realisations.

Date Created: 29/8/2003
Creator : Stampoultzis Michael

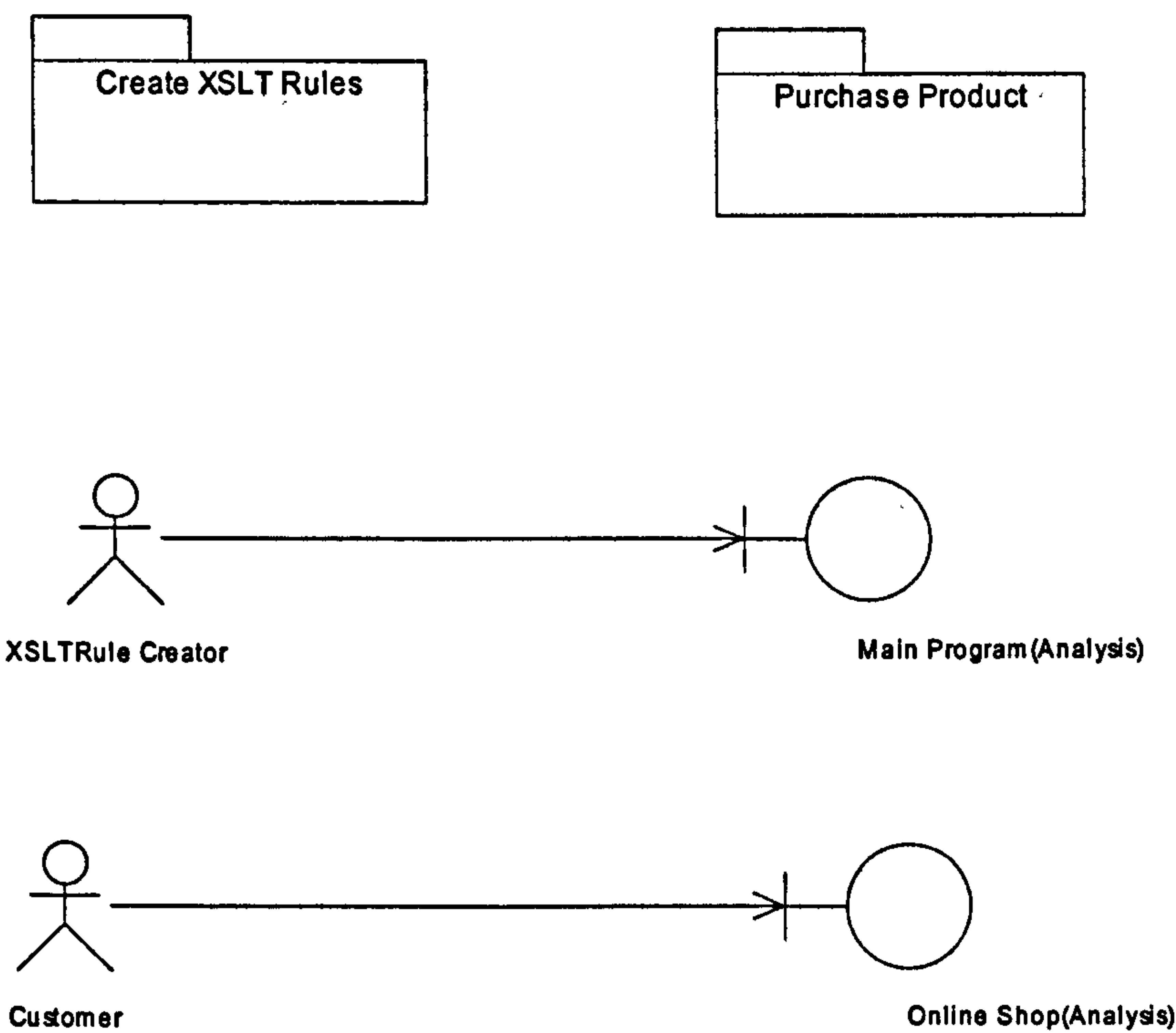


Figure 2: Package Overview (Analysis Model- Top Level Package)

Create XSLT Rules Use Case realization

These conceptual classes(also called "Analysis classes" are participating in the Use Case realizations. They are to be included in the Class Diagrams, Collaboration Diagrams and the Sequence Diagrams, realizing the Use Cases taken from the Use Case System Actually there is a "Trace" relationship between the Use Case System and the Analysis System. Now the Analysis System is a more detailed version of the Use Case System..

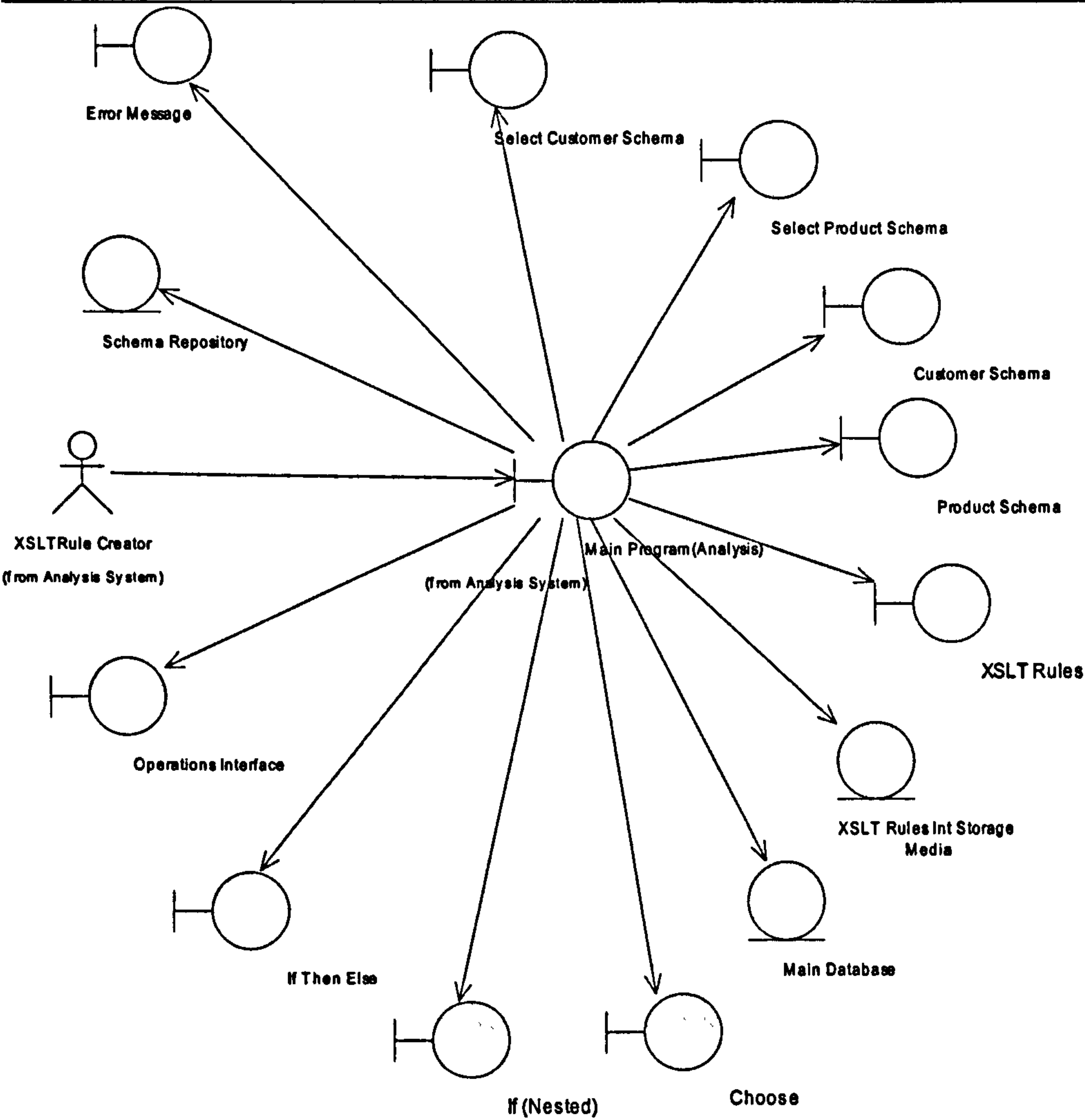
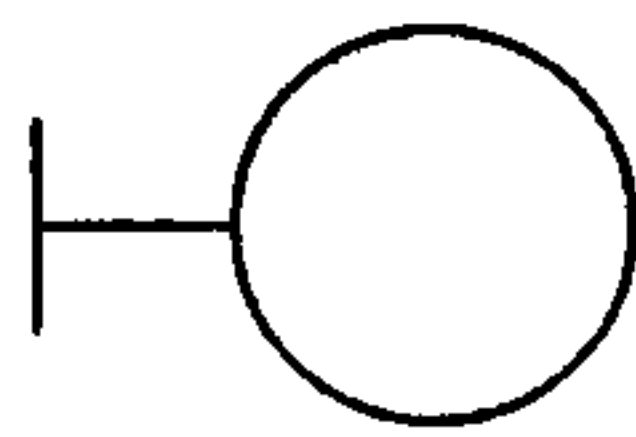


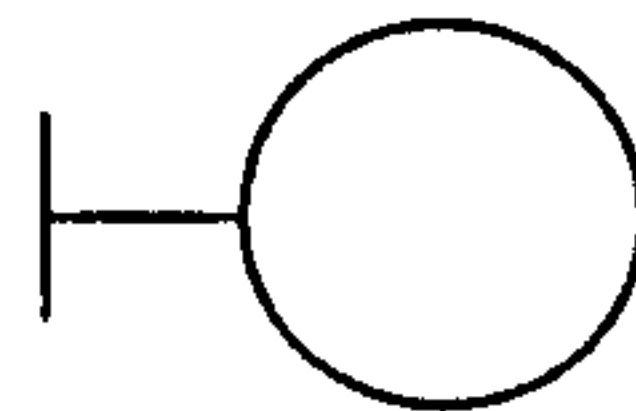
Figure 3: Create XSLT Rules Use Case realization (Package Overview)

Description of the analysis classes.



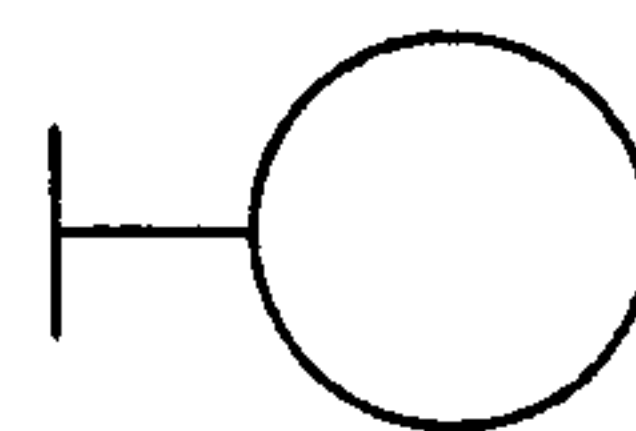
Select Product Schema

This is the interface which allows the XSLT Rules Creator actor to select the Product Schema.



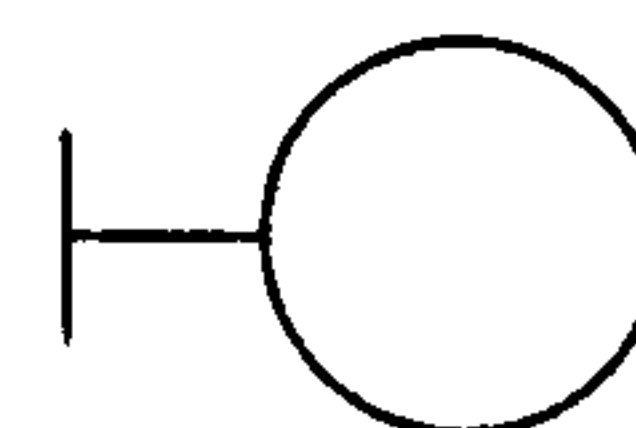
Select Customer Schema

This is the interface which allows the XSLT Rules Creator actor to select the Customer Schema.



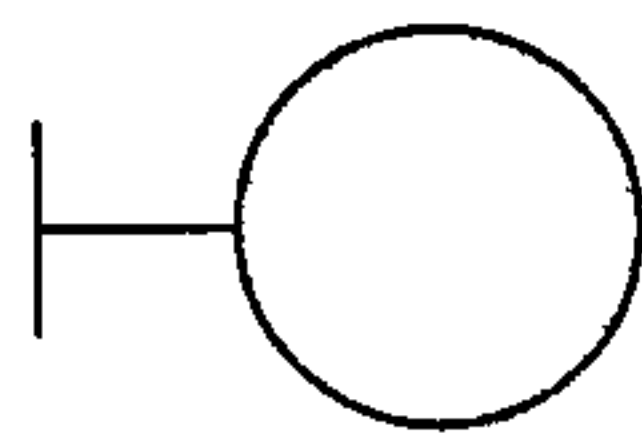
Error Message

If the Product Schema Name or the Customer Schema Name supplied whenever the XSLT Rules Creator actor uses the Select Product Schema or the Select Customer Schema interface classes then an error message is being produced.



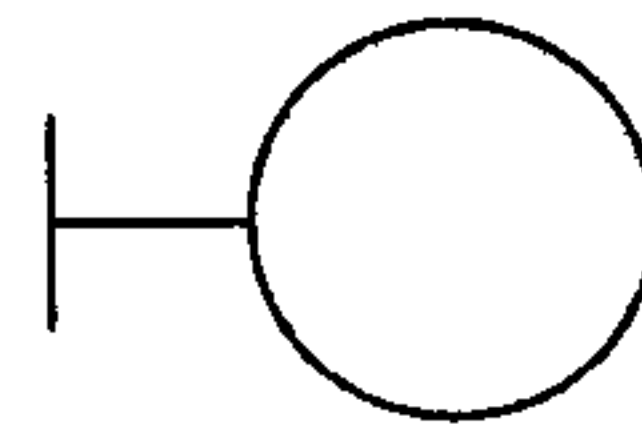
Customer Schema

This is the interface which displays the Customer Schema in a form of a Tree View.



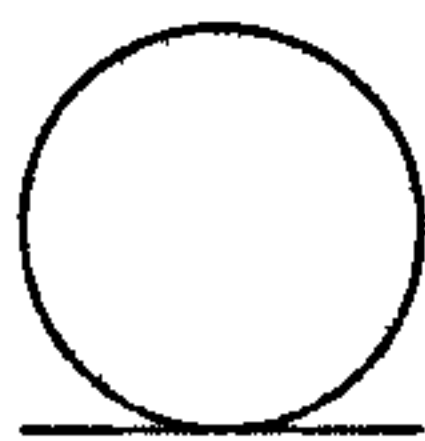
Product Schema

This is the interface which displays the Product Schema in a form of a Tree View.



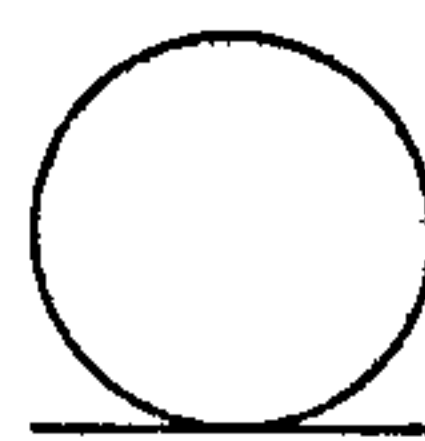
XSLT Rules

This is the interface which displays the XSLT Rules in a form of a Tree View.



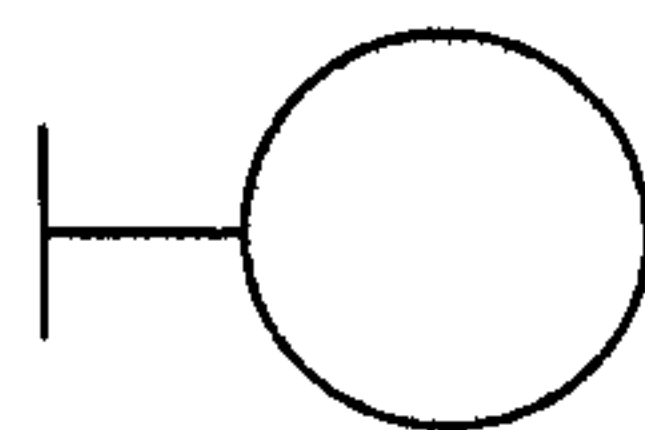
**XSLT Rules Int Storage
Media**

This is the entity Class responsible of storing-managing the XSLT Rules.



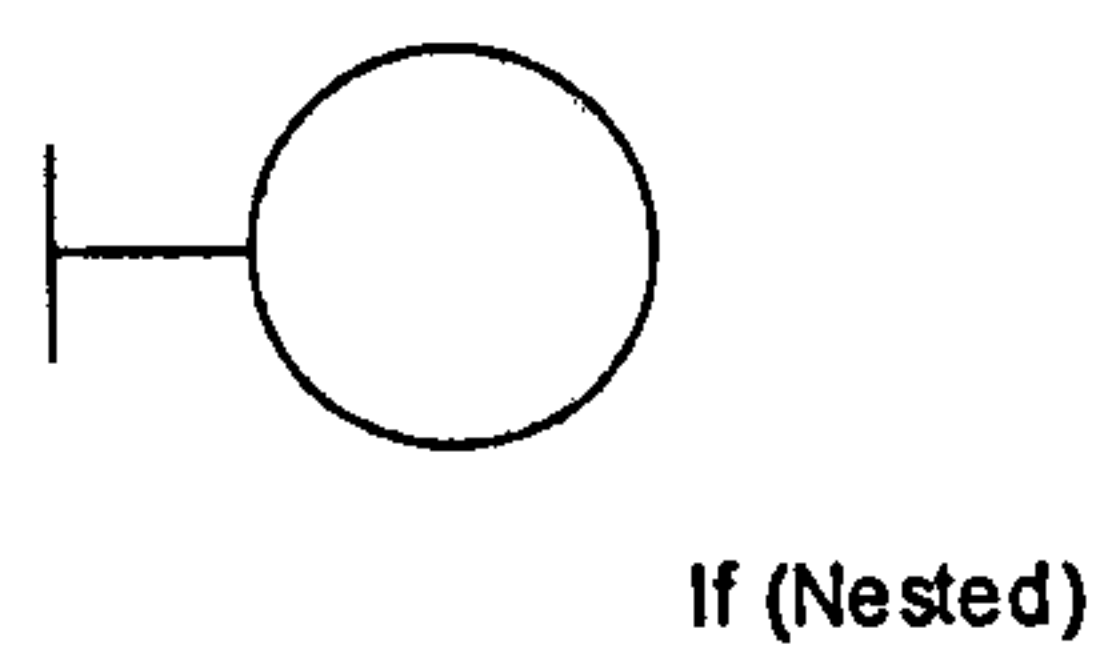
Main Database

This is the entity Class responsible of storing-managing permanently the XSLT Rules.

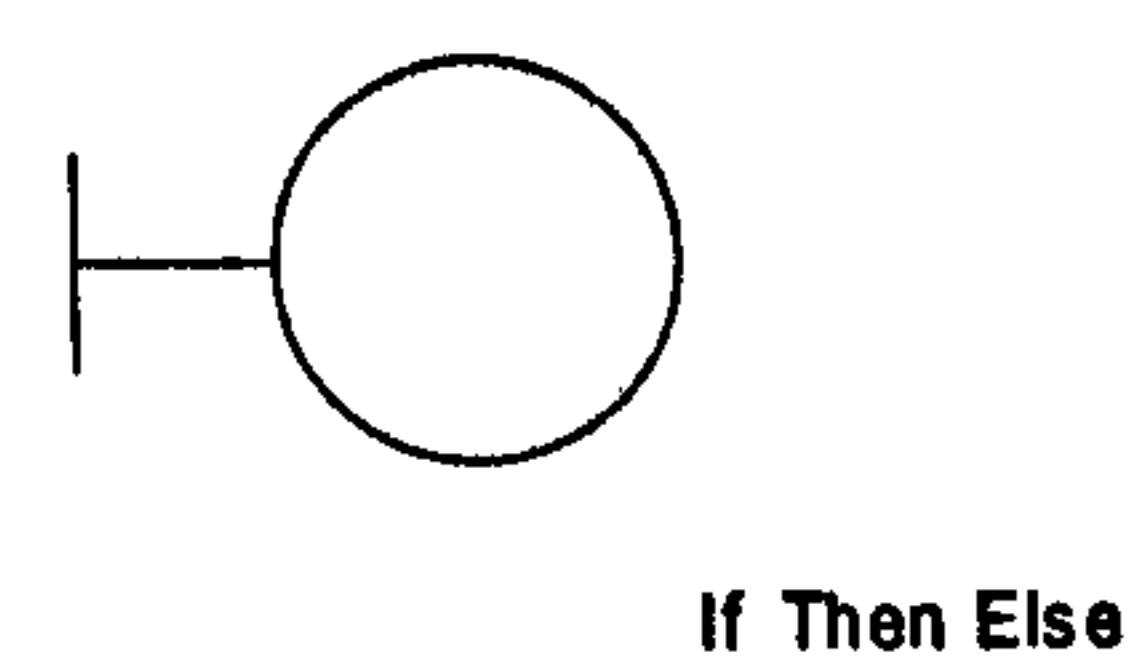


Choose

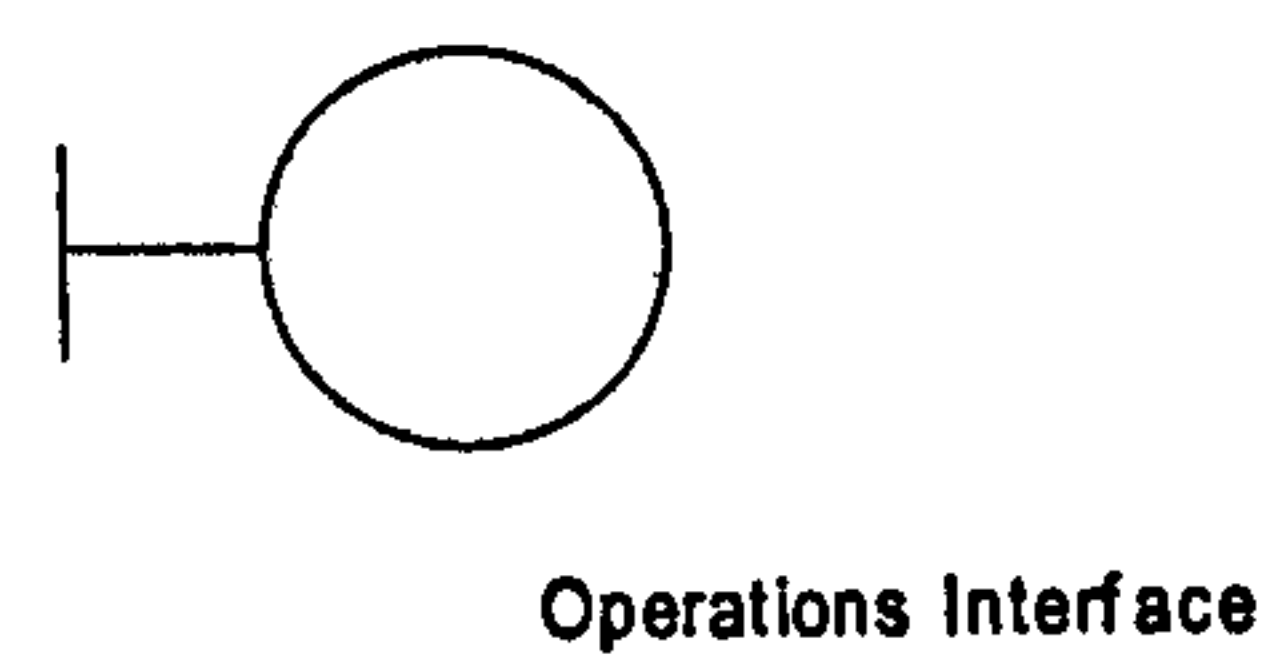
This is the interface which allows the XSLT Rule creator actor to create part of the XSLT rules.



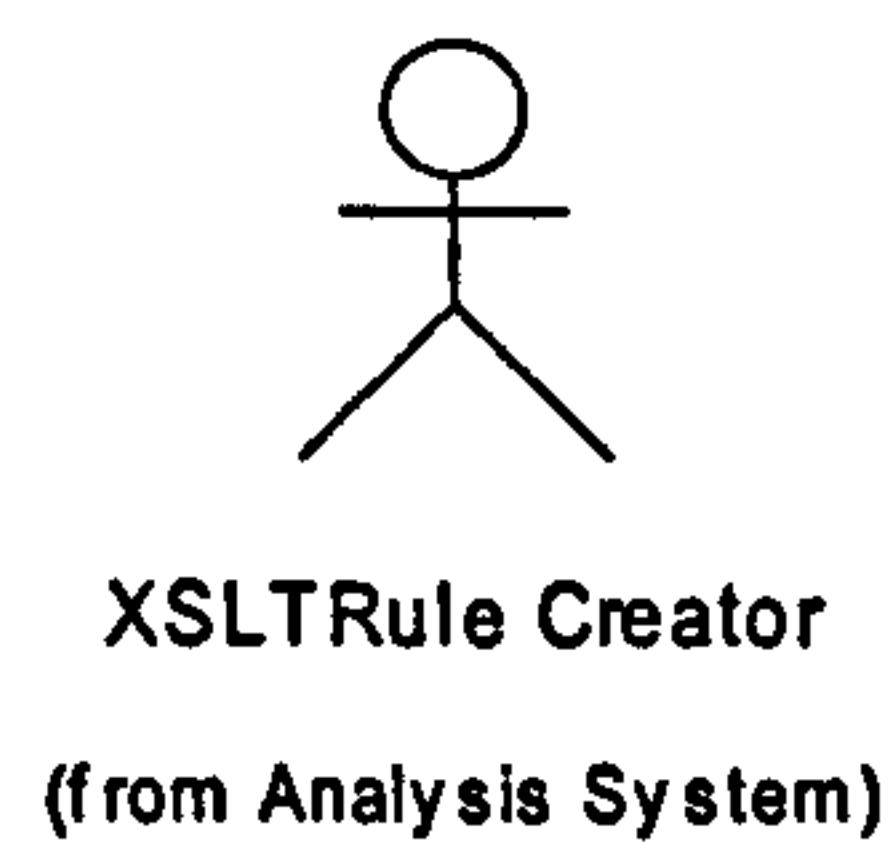
This is the interface which allows the XSLT Rule creator actor to create part of the XSLT rules.



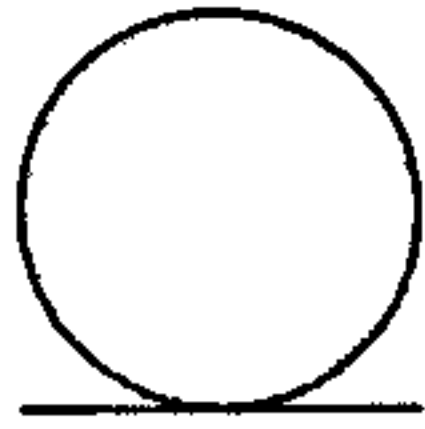
This is the interface which allows the XSLT Rule creator actor to create part of the XSLT rules.



This is the interface which accommodates and initiates the Choose, If(Nested), If Then Else interfaces.



This actor is actually initialises the use case, and creates or modifies and then saves the XSLT Rules.



Schema Repository

This is the entity Class which is being used to store the Product Category Schemas and the Customer Segment Schemas.

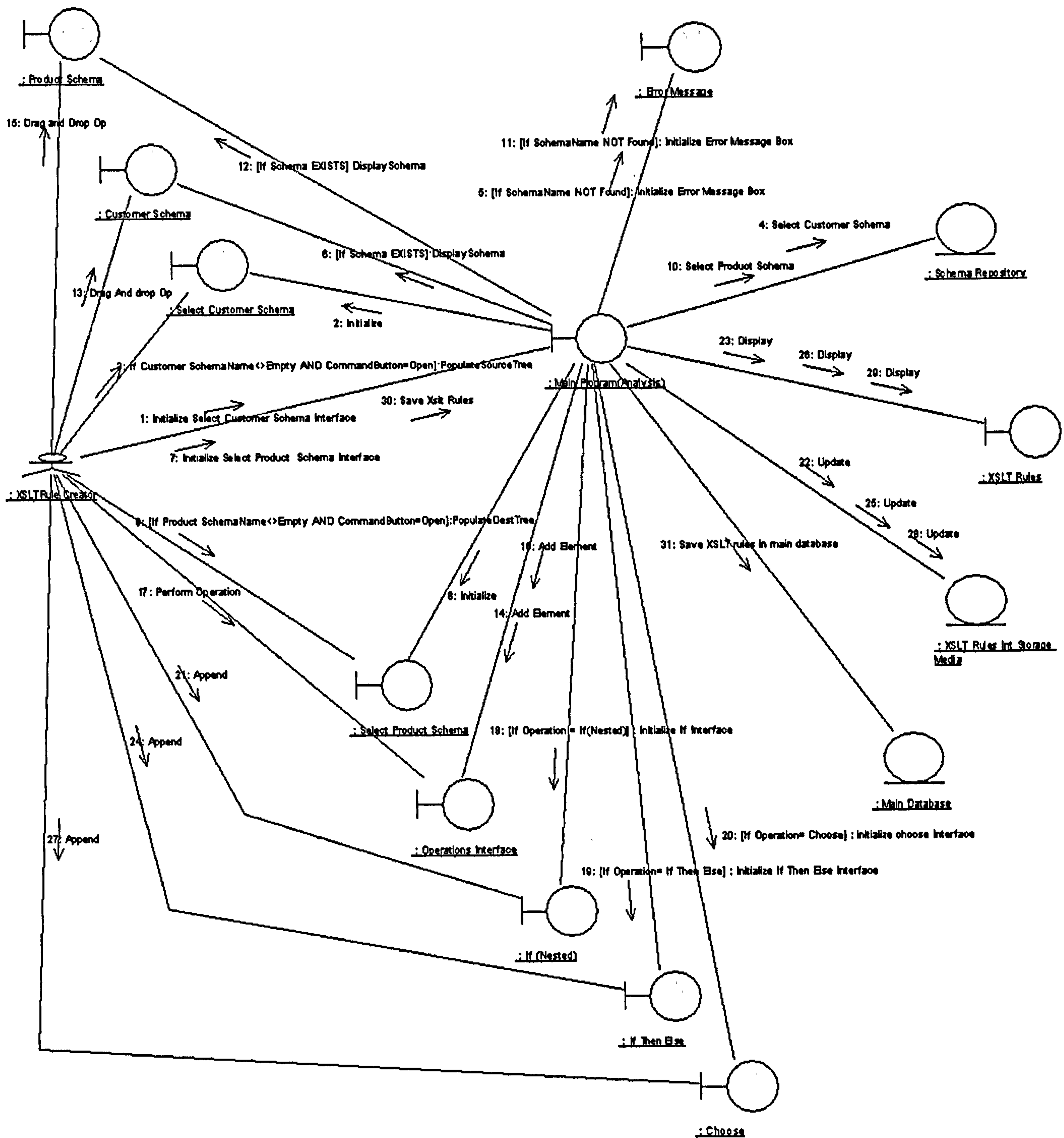


Figure 4: Create XSLT Rules Collaboration Diagram.

Flow of Events for Create XSLT Rules Collaboration diagram

The XSLT Rules Creator actor selects (1) *CustomerSchema* action button from the Main Interface.

The Main Program then causes the initialization of the Select Customer Schema Interface class by calling the (2) Initialize method.

If Customer Schema Name Is Not Empty and the Command Button on the Customer Schema Interface is selected then Populate the Source Tree.

(3). If Customer SchemaName \neq Empty AND CommandButton=Open]: PopulateSourceTree.

(4). Select Customer Schema. This action is issued from the main Program to the Schema Repository entity class.

(5). [If SchemaName NOT Found]: Initialize Error Message Box.

(6). [If Schema EXISTS]: Display Schema. Then the XSLT Rules Creator actor selects Product Schema action button from the Main MDI (Multiple Document Interface) Interface. (7). Initialize Select Product Schema Interface.

The Main Program then causes the initialization of the Select Product Schema Interface class by calling the (8) Initialize method. If Product Schema Name Is Not Empty and the Command Button on the Product Schema Interface is selected then Populate the Source Tree.

(9). If Product SchemaName \neq Empty AND CommandButton=Open]: PopulateSourceTree.

(10). Select Product Schema. This action is issued from the main Program to the Schema Repository entity class.

(11). [If SchemaName NOT Found]: Initialize Error Message Box.

(12). [If Schema EXISTS]: Display Schema.

The XSLT Rules Creator actor then initializes a Drag and Drop Operation. From the Customer Schema Interface class to the Operations Interface Class. (13). Drag And

drop Op. Then an element from the Customer Schema is being added to the Interfaces interface class. (14). Add Element. The XSLT Rules Creator actor then initializes a Drag and Drop Operation. From the Product Schema Interface class to the Operations Interface Class.

(15). Drag and Drop Op. Then an element from the Product Schema is being added to the Interfaces interface class. (16). Add Element. Then the actor initiates an operation within the Operations interface class. (17). Perform Operation.

If the Operation selected is: If (Nested) then initialize the If Interface. (18). [If Operation = If(Nested)] : Initialize If Interface.

If the Operation selected is: If Then Else then initialize the If Then else Interface. (19). [If Operation= If Then Else]: Initialize If Then Else Interface.

If the operation selected is: choose then select the Choose Interface. (20). [If Operation= Choose]: initialize choose Interface.

If the If(Nested) interface was selected at first time and the rules were created then the next step is to select the Append Button from this interface in order to append the XSLT rules fragment into the main body of the XSLT rules. (21). Append. Then this change in the main body of the XSLT rules is being reflected by calling the (22). Update and then the (23). Display messages.

If the If Then Else interface was selected at first time and the rules were created then the next step is to select the Append Button from this interface in order to append the XSLT rules fragment into the main body of the XSLT rules. (24). Append. Then this change in the main body of the XSLT rules is being reflected by calling the (25). Update and then the (26). Display messages.

If the Choose interface was selected at first time and the rules were created then the next step is to select the Append Button from this interface in order to append the XSLT rules fragment into the main body of the XSLT rules. (27). Append. Then this change in the main body of the XSLT rules is being reflected by calling the (28). Update and then the (29). Display messages.

The XSLT Rules actor then selects the SaveXSLT Command Button from the Main Interface. (30). SaveXSLT. Then the Main Database saves the XSLT Rules (31). Save XSLT rules in main database.

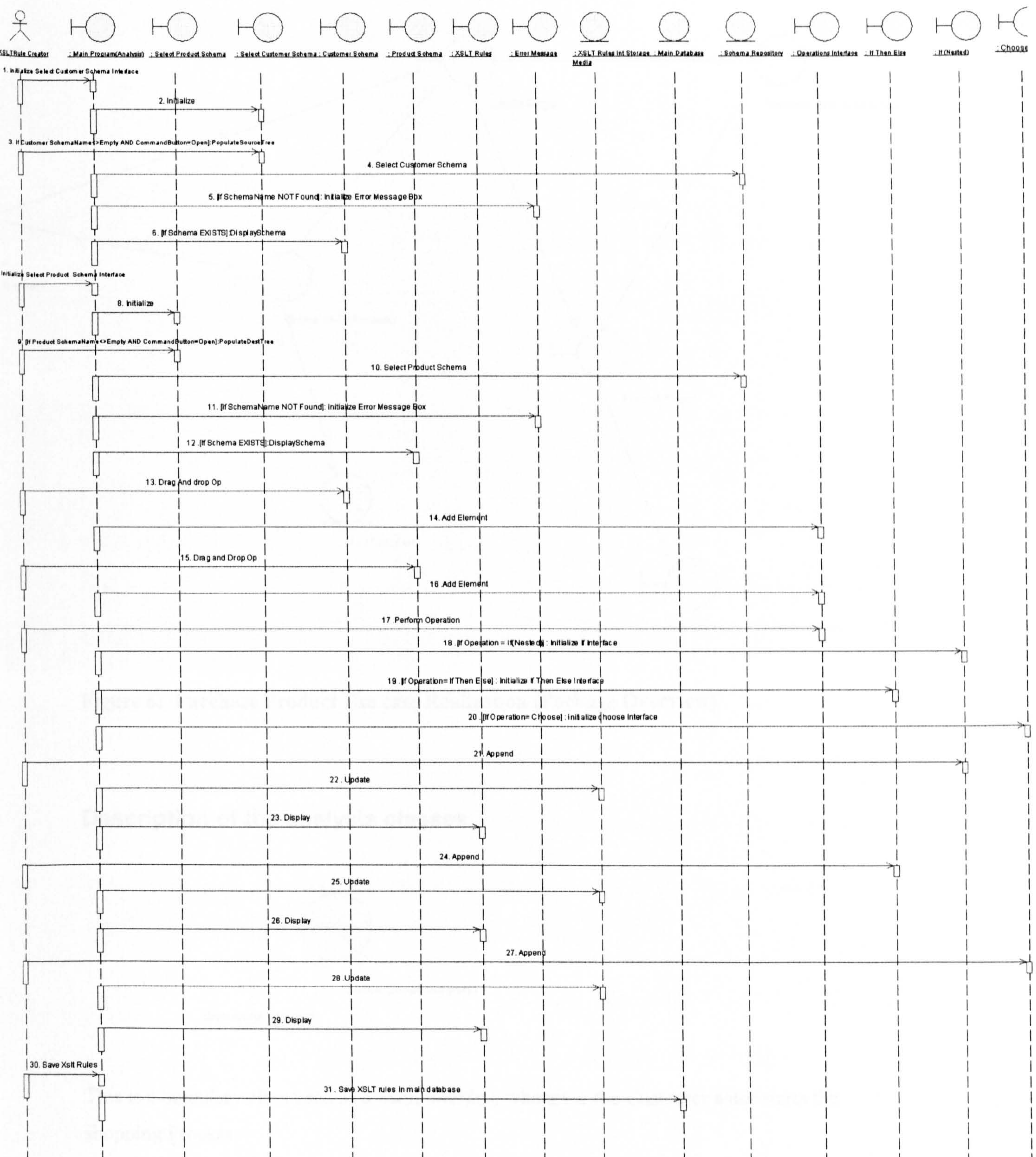


Figure 5: Create XSLT Rules Sequence Diagram

Purchase Product Use Case Realization

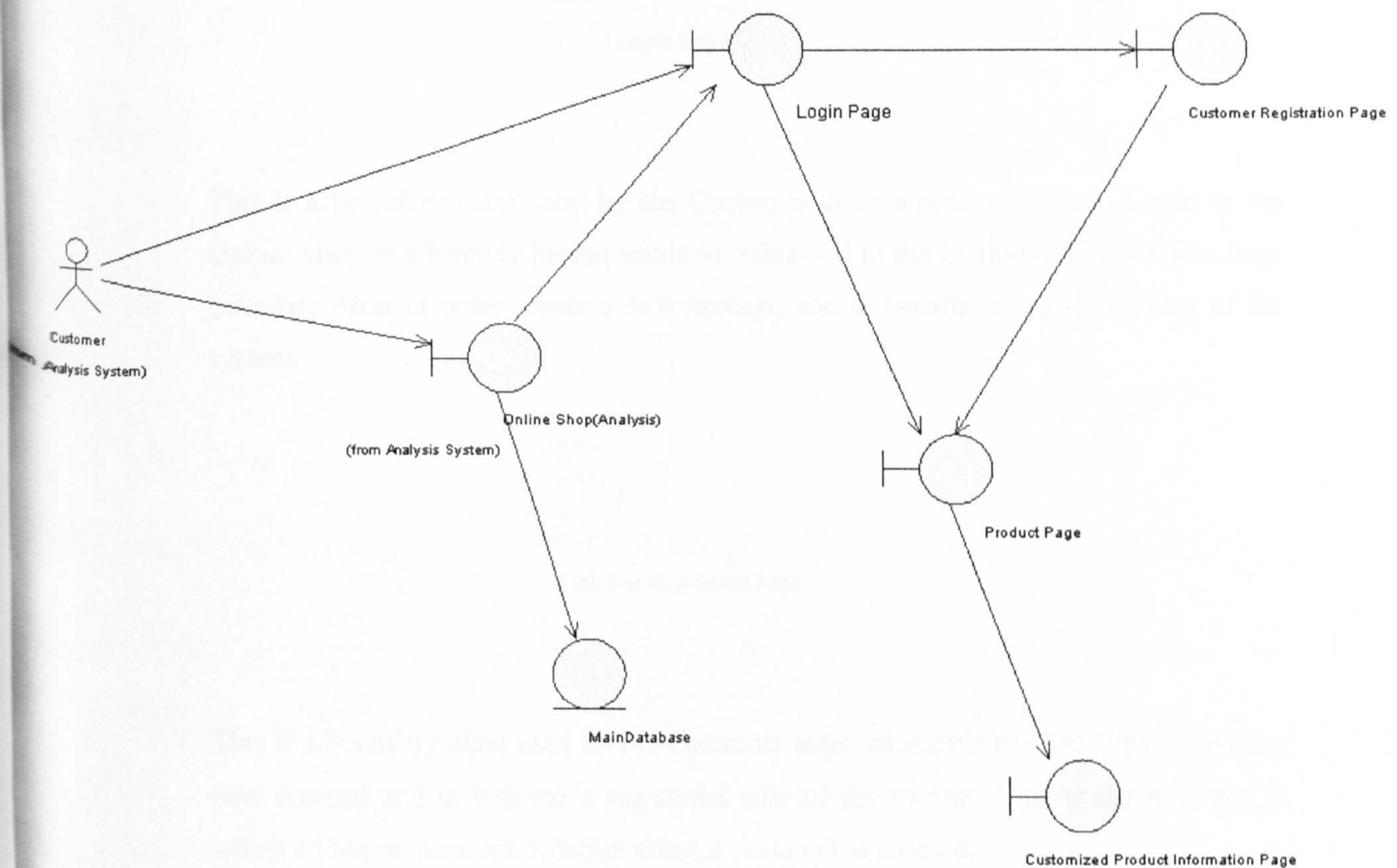
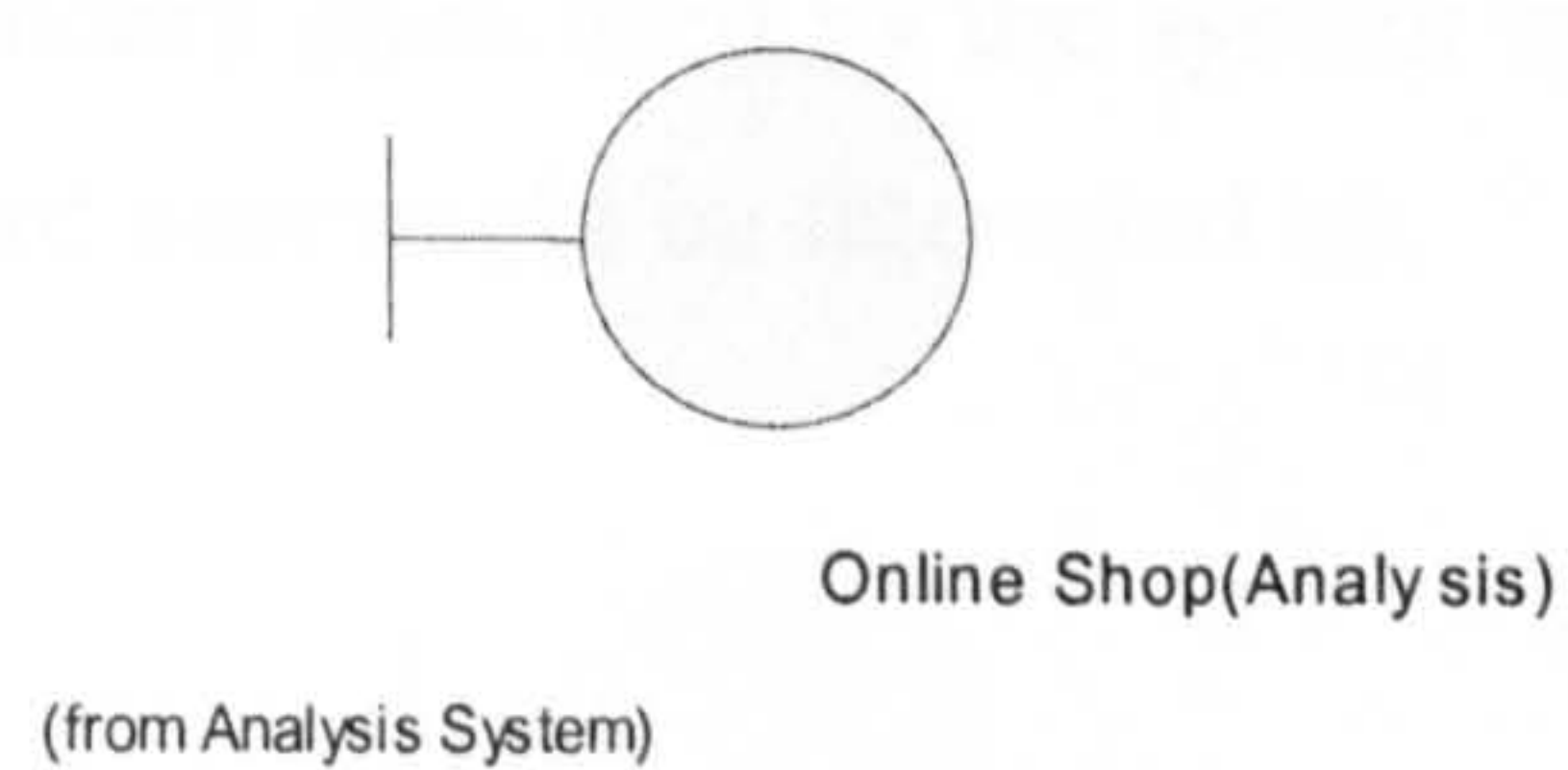
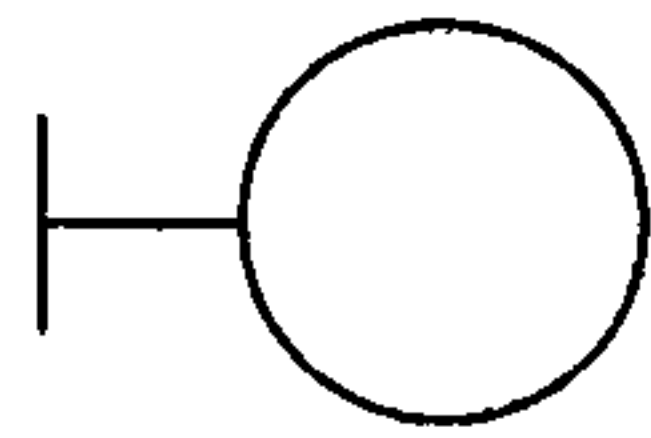


Figure 6: Purchase Product Use case Realization (Package Overview)

Description of the analysis classes.

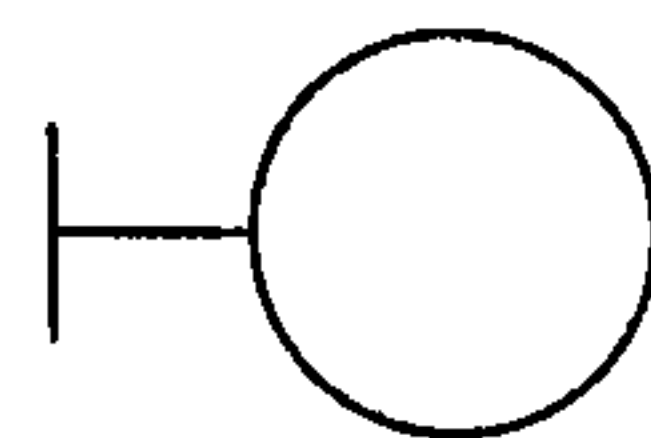


This is a boundary class used as a main interface whenever the Customer actor starts the shopping Process.



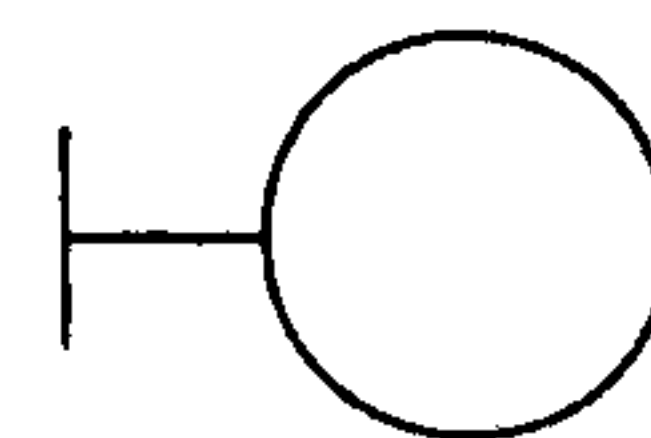
Login Page

This is a boundary class used by the Customer actor whenever tries to Login in the Online Shop or whenever he/she wants to redirected to the Customer Registration Page boundary class in order create a new account and to became a registered user of the system.



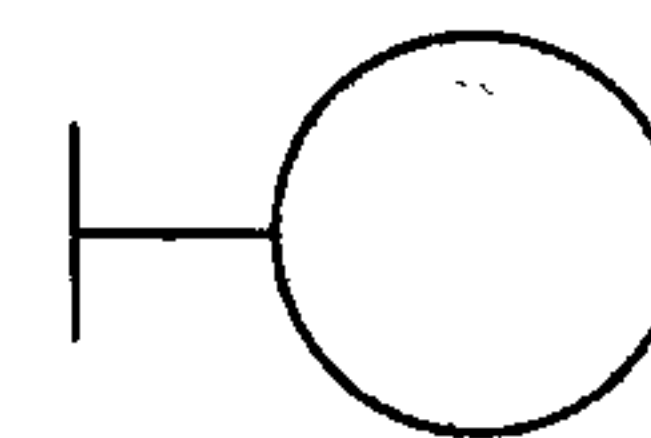
Customer Registration Page

This is a boundary class used by the Customer actor whenever he/she wants to create a new account and to become a registered user of the system. This is the interface in which all the relevant information about a customer is entered.



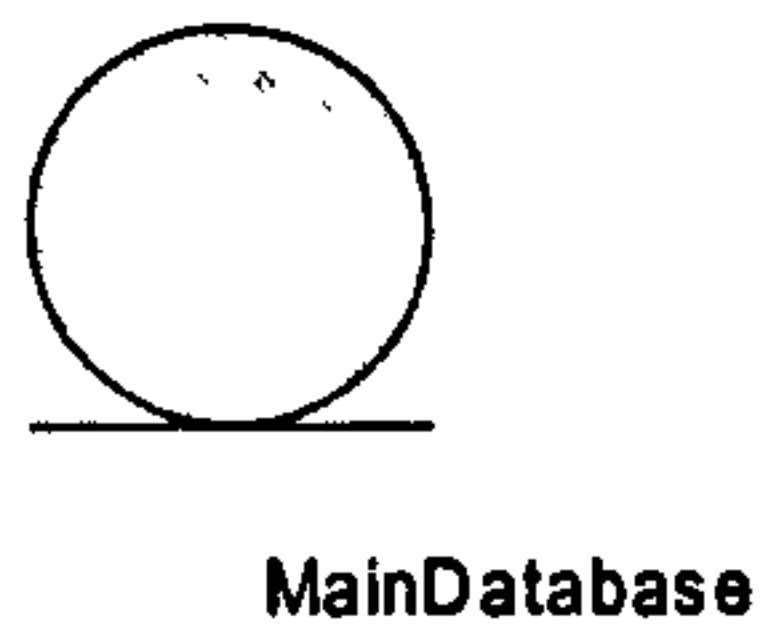
Product Page

This is a boundary class used by the system in order to display all the relevant products that the current user might be interested on.

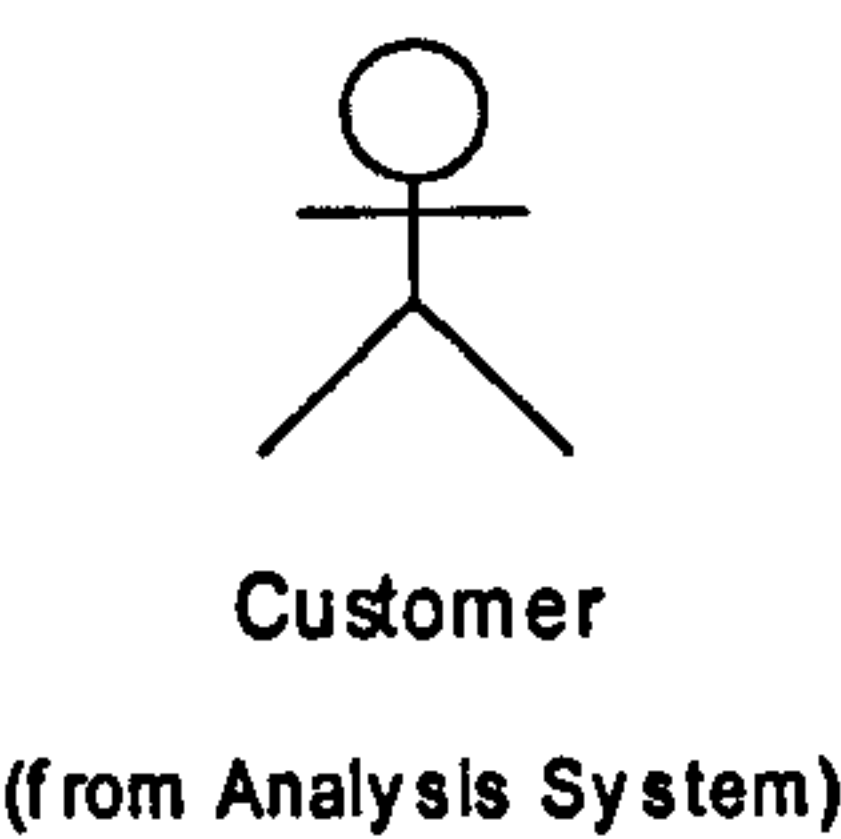


Customized Product Information Page

This is a boundary class used by the system in order to display all the relevant information about a specific product based on the customer's profile.



This is an entity class which is being used by the system in order to store information about the customer profile, customer segments, product information, and also the XSLT Rules for a specific customer segment and some specific product categories.



This actor is actually initialises the use case, and purchases products.

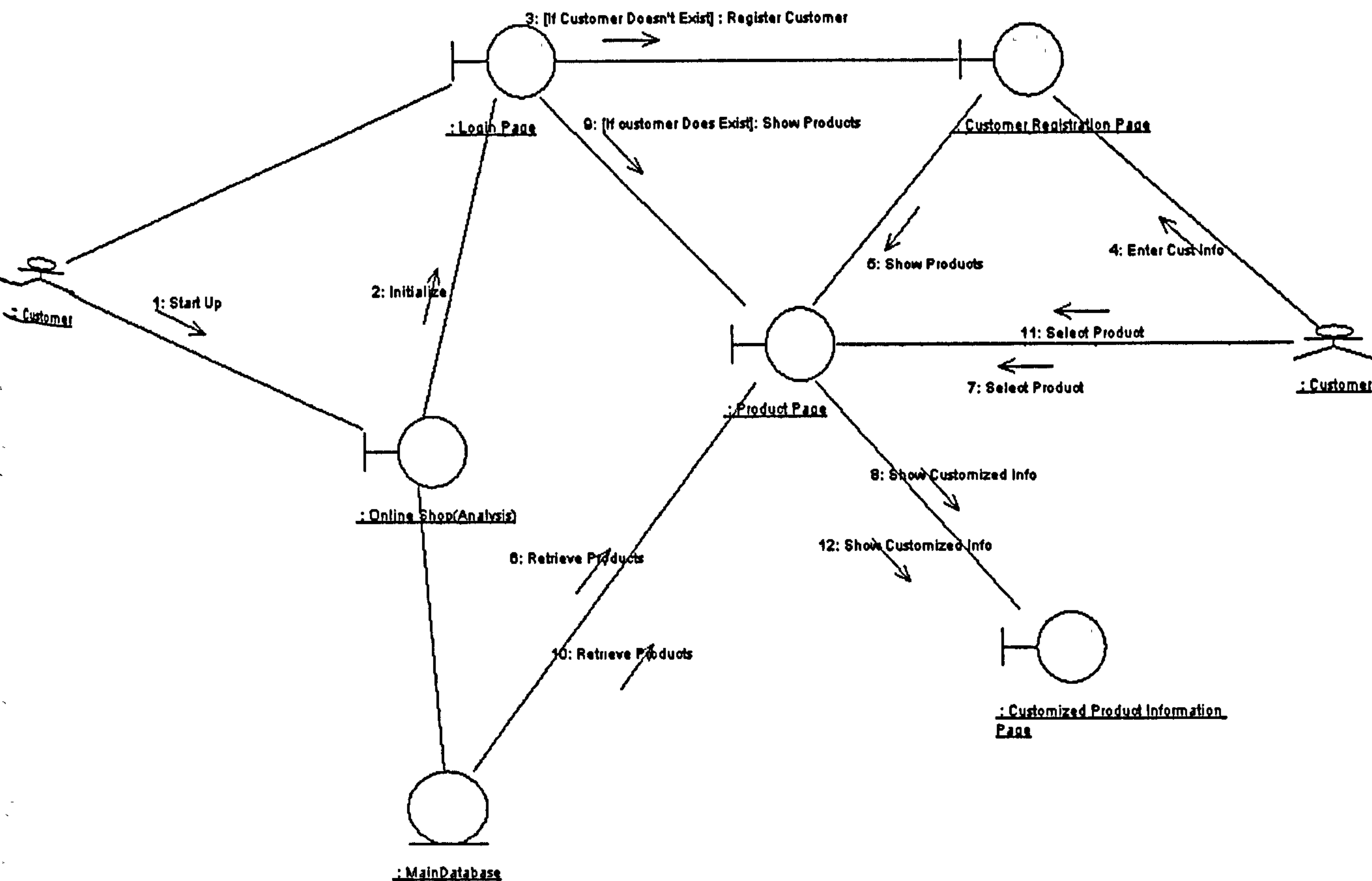


Figure 7: Purchase Product Collaboration Diagram.

Flow of events for the “Purchase Product” Collaboration Diagram.

The Customer initializes the application (Online Shop). (1). Start Up is called and the Online shop application starts up. (2). Initialize message is sent to the Login Page which is being displayed. A test is being performed in order to test whether the customer exists or not. If the Customer doesn't exist the customer is redirected to the Customer Registration Page. (3). [If Customer Doesn't Exist]: Register Customer. Then the customer enters his/hers personal information. (4). Enter Cust Info. When the registration is finished the Customer is redirected to the Product page. (5). Show Products.

The information for the products is being retrieved from the Main database. (6).Retrieve Products. Then the Customer can make a selection from the product list. (7). Select product. The customized information for a specific product is being displayed to a different page. (8). Show Customized Info. In the other hand if the customer exists then is being redirected to the product page whenever he/she logins. (9). [If customer Does Exist]: Show Products. The information for the products is being retrieved from the Main database. (10).Retrieve Products. Then the Customer can make a selection from the product list. (11). Select product. The customized information for a specific product is being displayed to a different page. (12). Show Customized Info.

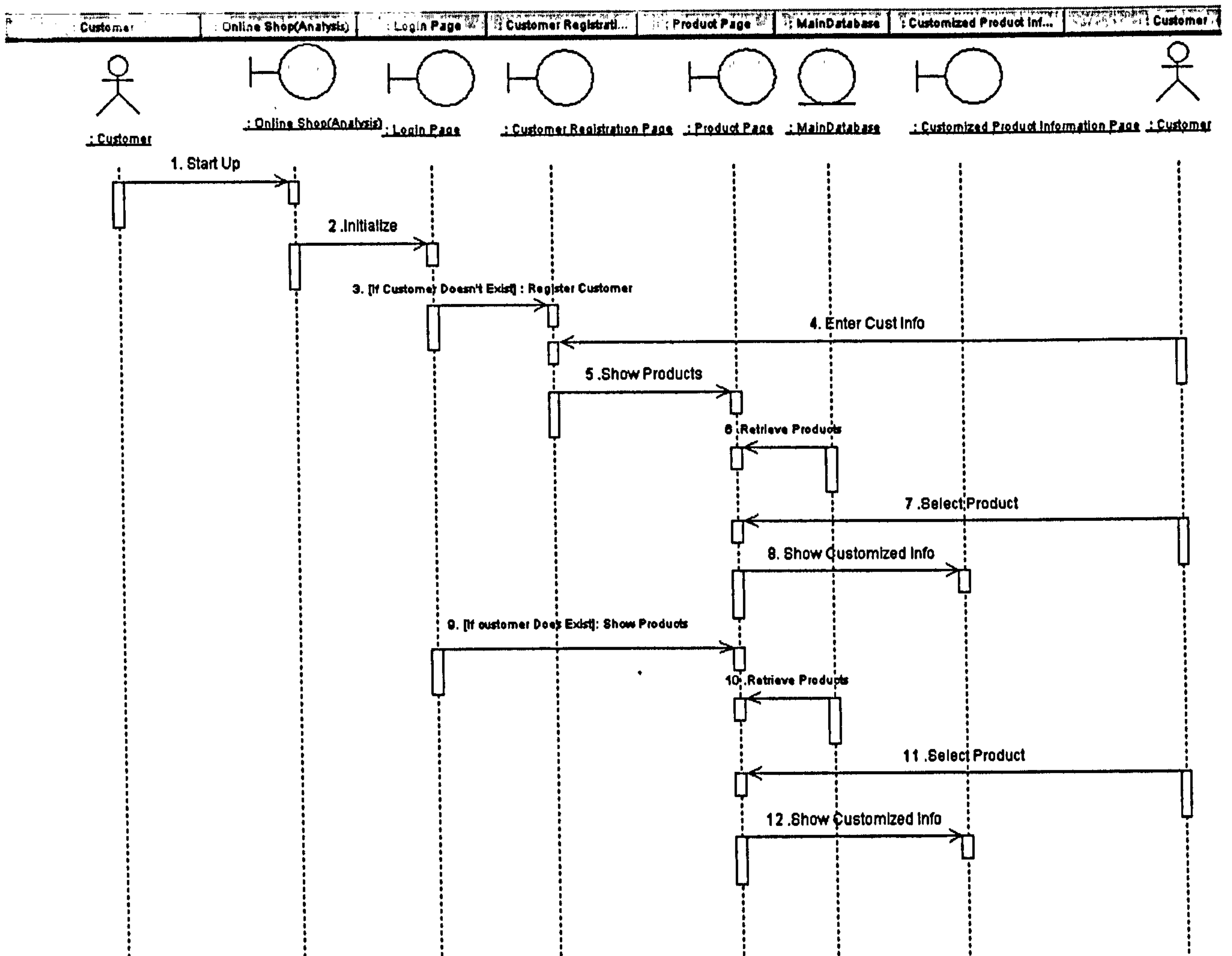


Figure 8: Purchase Product Sequence Diagram.

4.3 Design Model

The Design Model actually contains the Design System that we are describing here.

The Design Model is an object model that describes the physical realization of use cases by focusing on how functional and no functional requirements, together with other constraints related to the implementation environment, impact the system under consideration. In addition, the Design Model serves as an abstraction on the system's implementation and is thereby used as an essential input to activities in implementation.

The Design Model is represented by a Design System that denotes the top-level subsystem of the model.

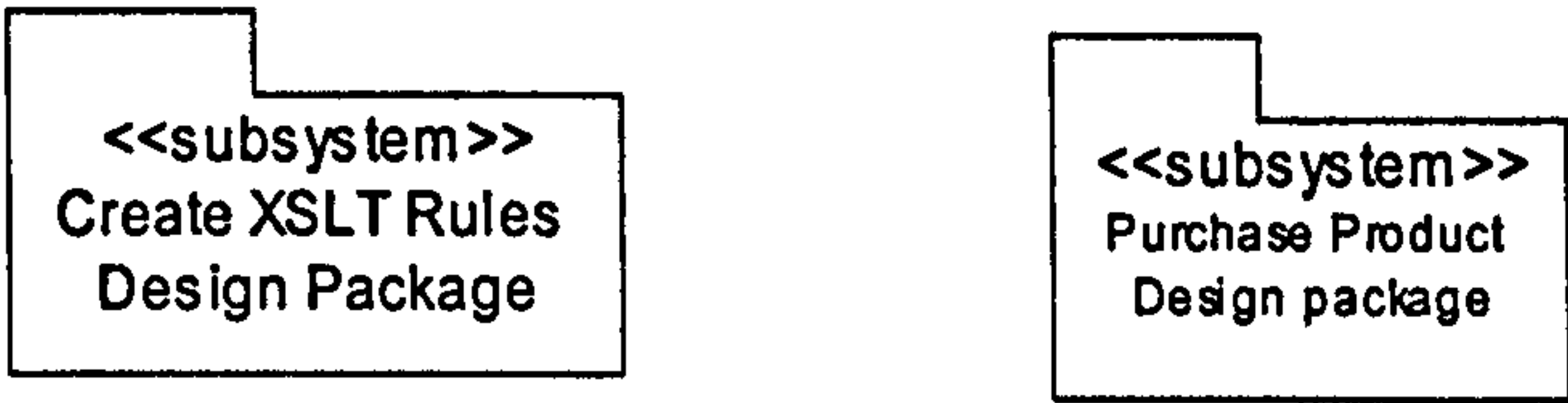
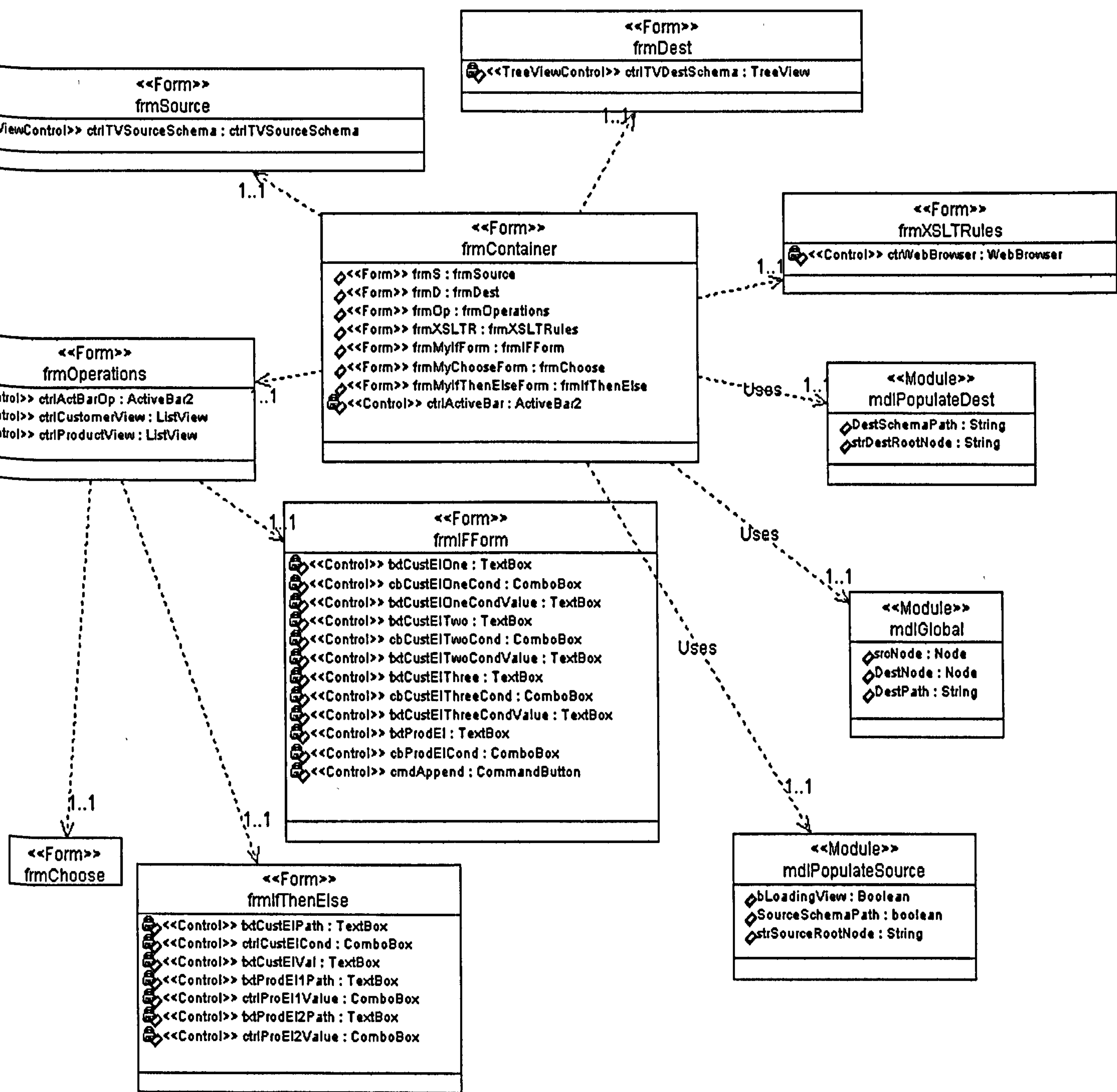


Figure 9: Design Model (Package Overview)

Create XSLT Rules Design Package (Subsystem)



10: Create XSLT Rules Package Overview (Design Model)

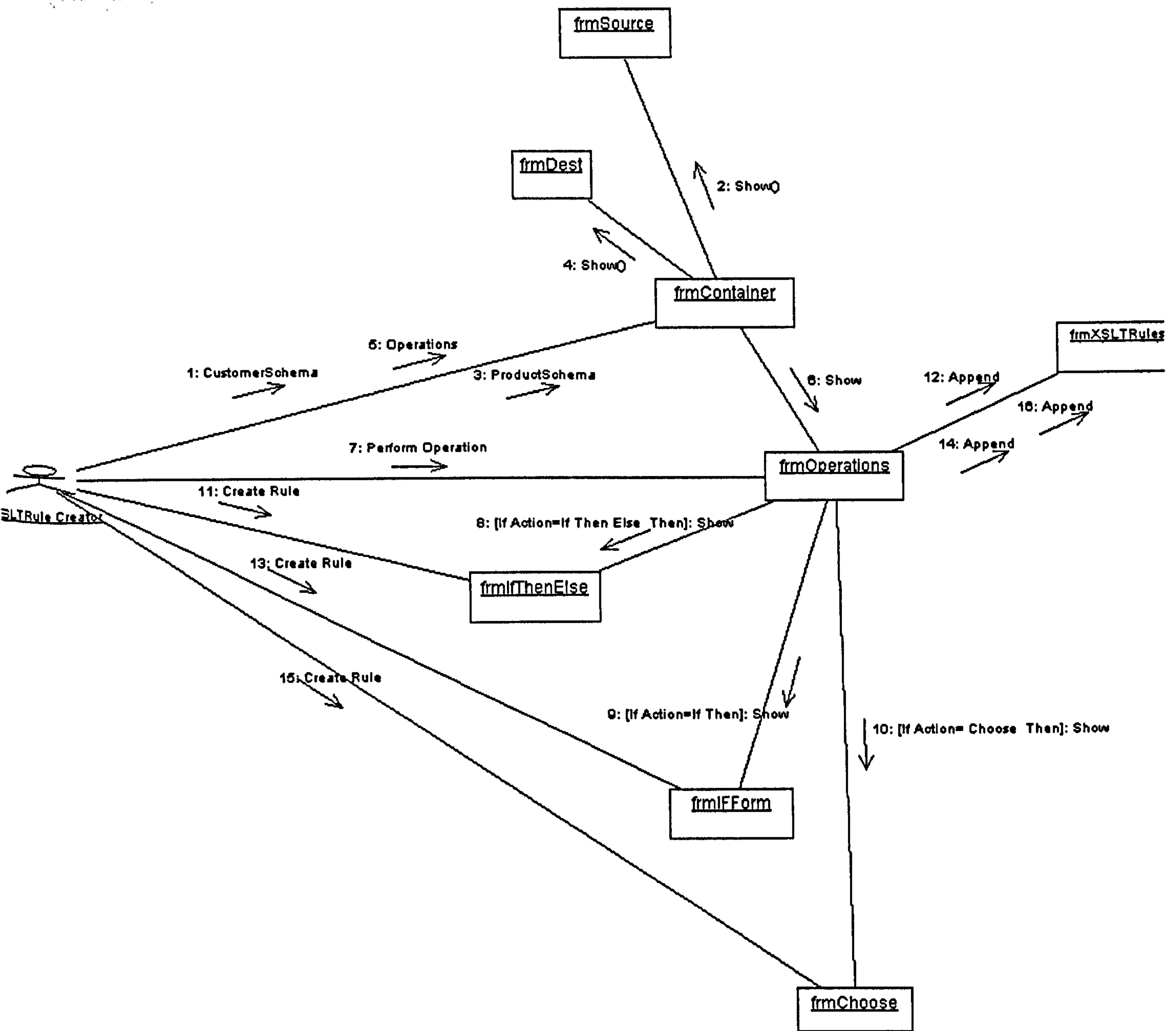


Figure 11: Create XSLT Rules Collaboration Diagram (Design Model)

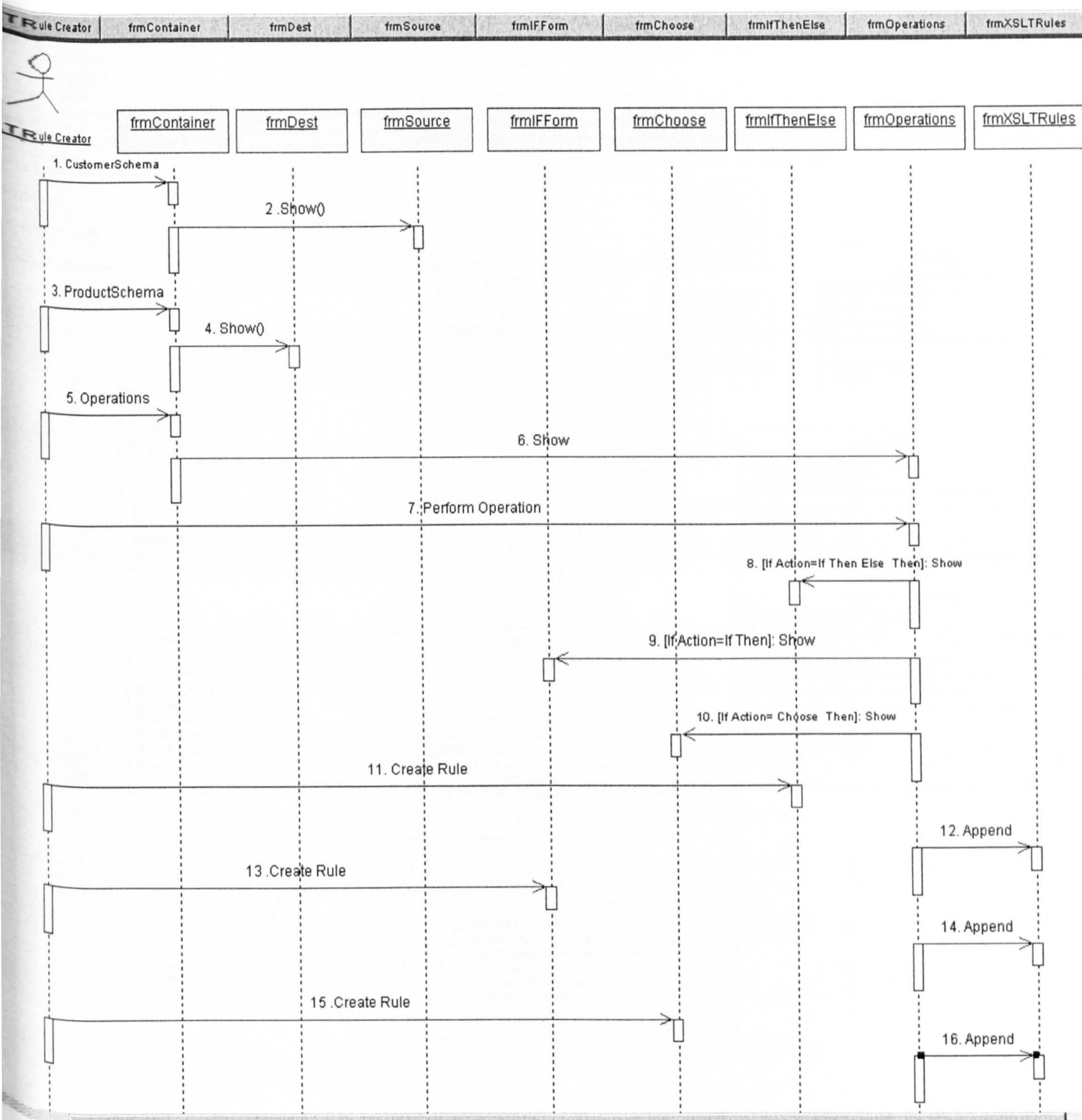


Figure 12: Create XSLT Rules Sequence Diagram (Design Model)

Purchase Product Design package (Subsystem)

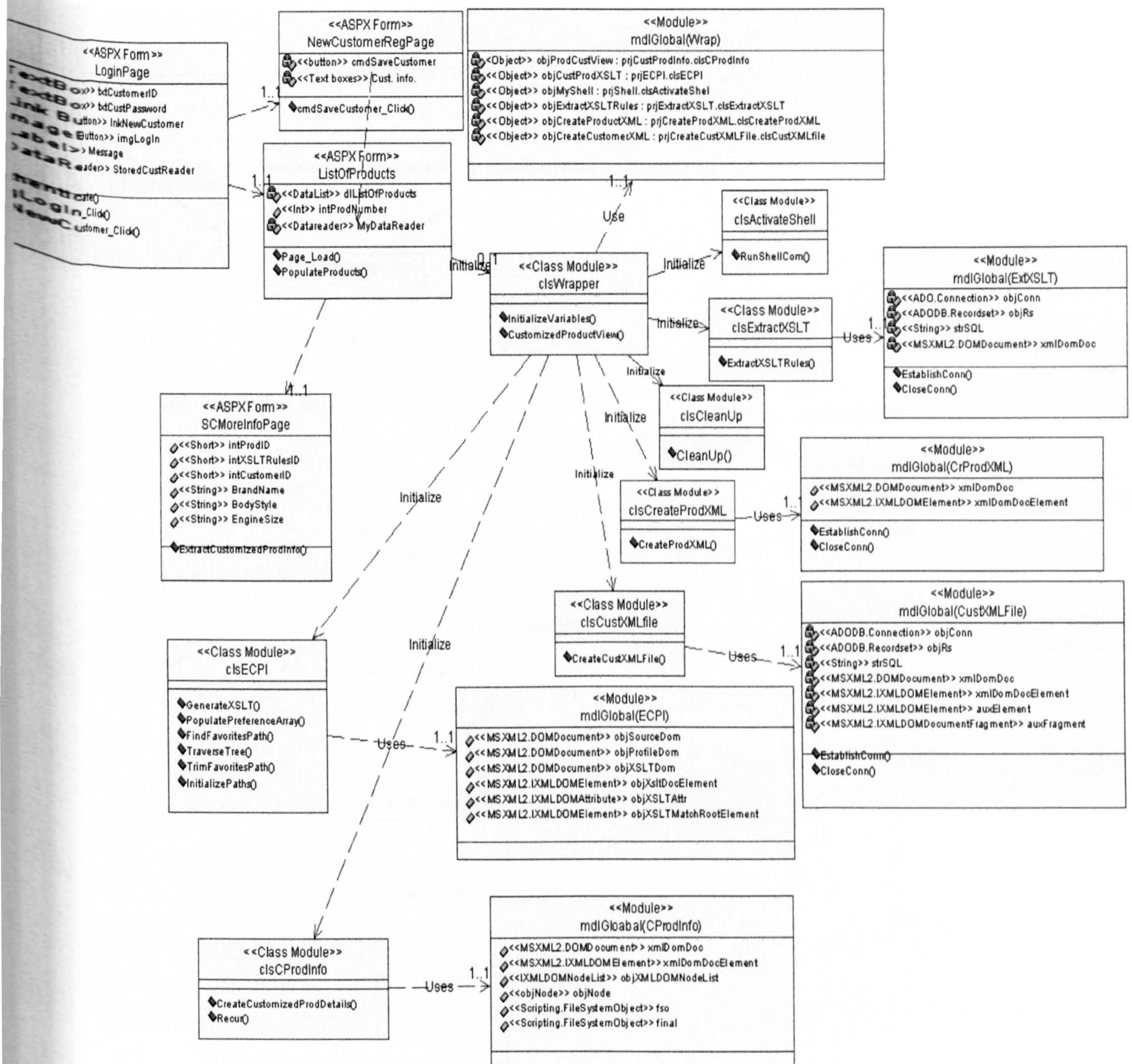


Figure 13: Purchase Product Design (Package Overview)

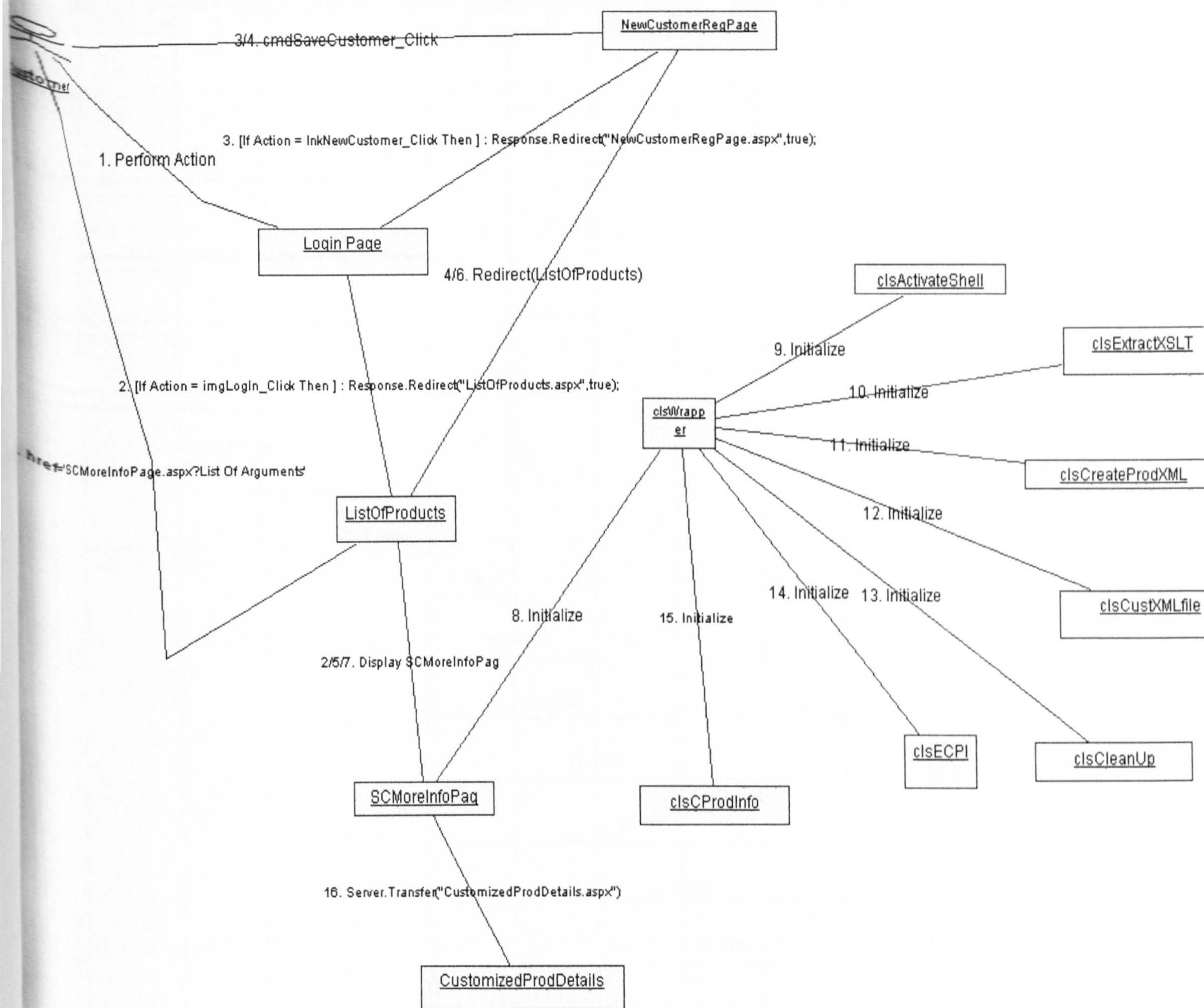


Figure 14: Purchase Product Collaboration Diagram (Design Model)

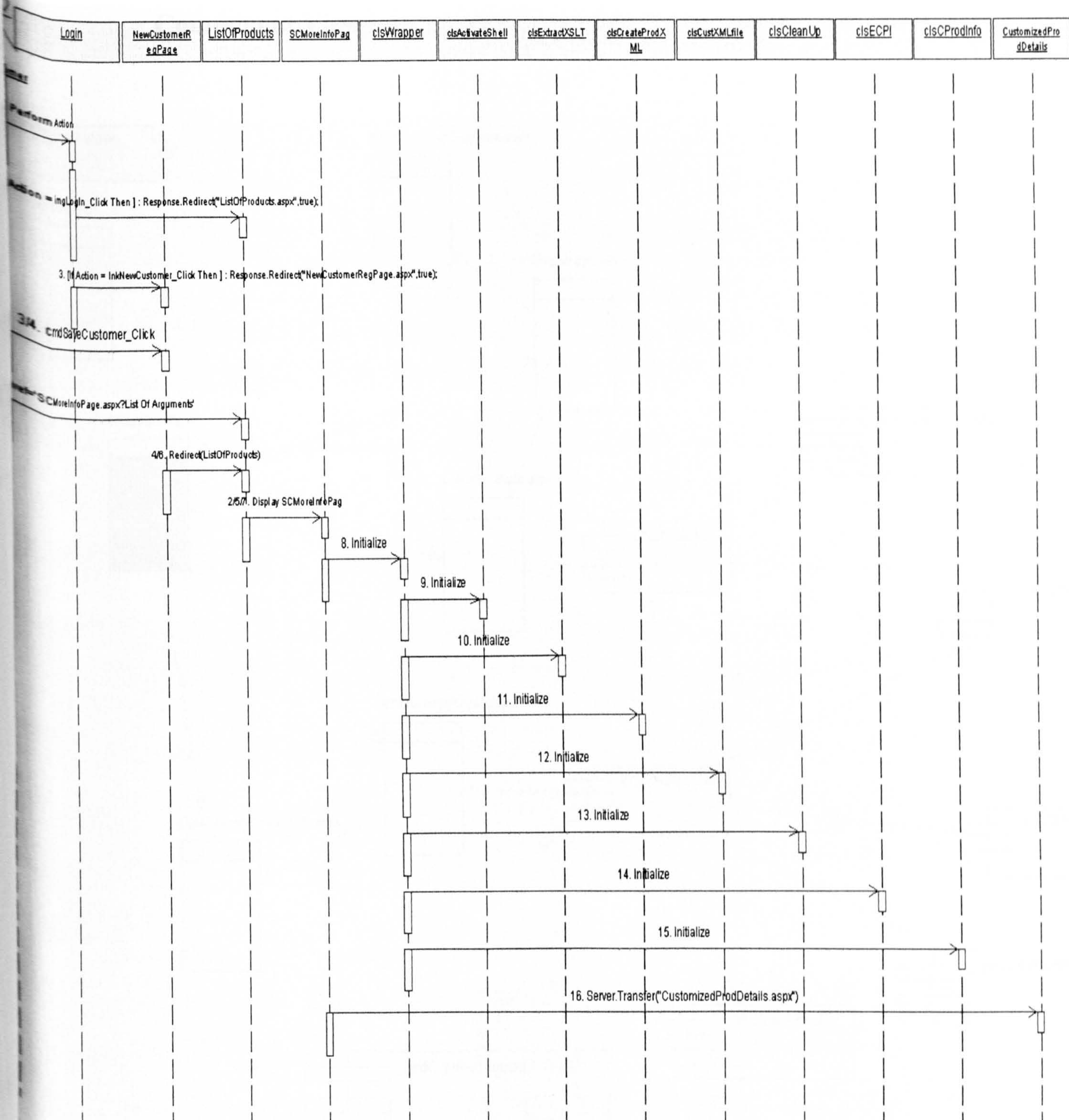


Figure 15: Purchase Product Sequence Diagram (Design Model)

Component View

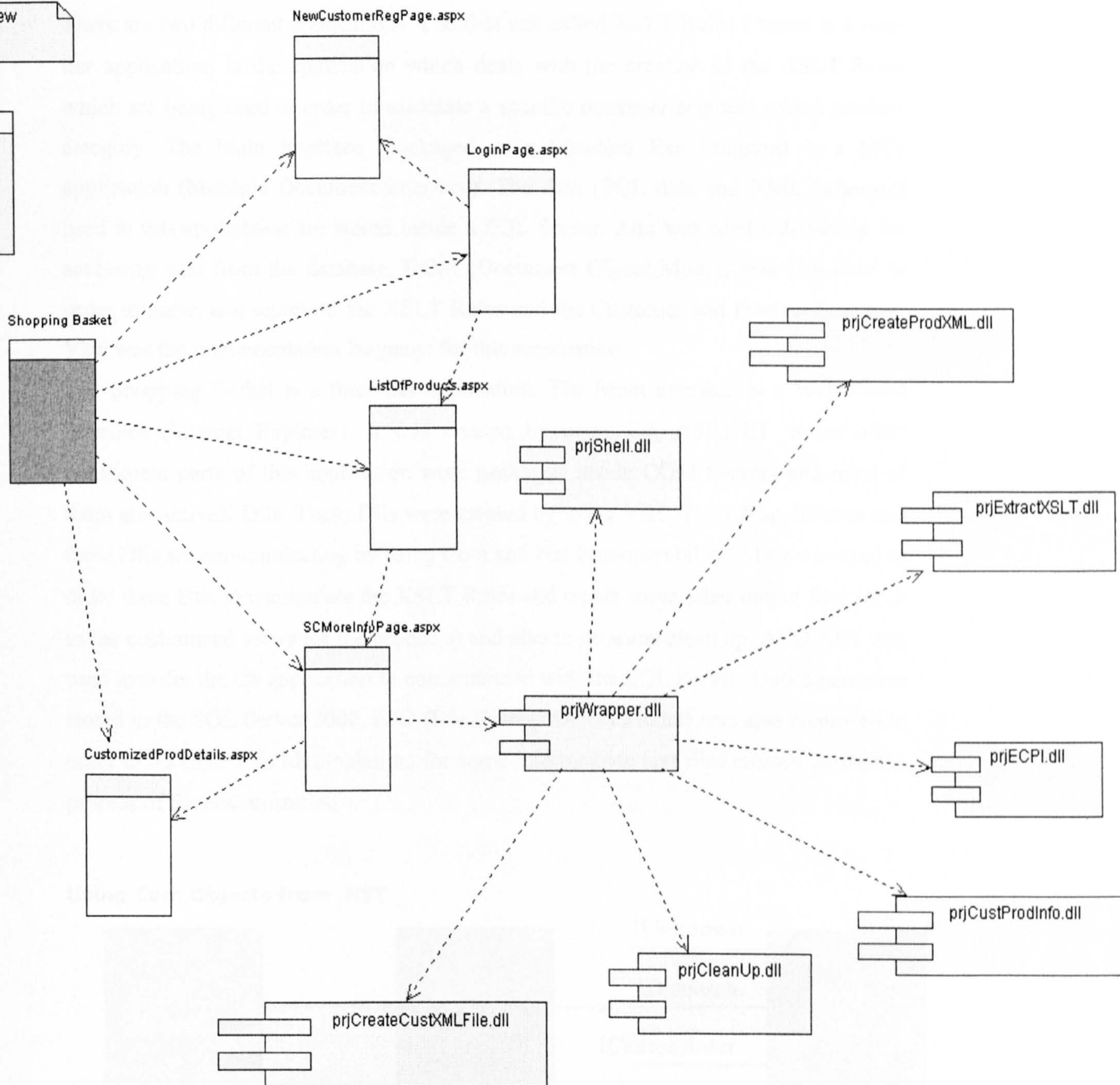


Figure 16: Component View

4.4 Implementation Approach

There are two different applications. The first one called XSLT Rules Creator is a two-tier application. Is the application which deals with the creation of the XSLT Rules which are being used in order to associate a specific customer segment with a product category. The Main Interface (packaged as a Standard Exe Program) is a MDI application (Multiple Document Interface). The data (SQL data and XML Schemas) used in this application are stored inside a SQL Server. Ado was used extensively for accessing data from the database. DOM (Document Object Model) was also used in order to parse, and represent the XSLT Rules and the Customer and Product Schemas. VB6 was the implementation language for this application.

The Shopping Basket is a three-tier application. The Main interface is a Web Based Interface (Internet Explorer). It was created by using C#, ASP.NET. Some other constituent parts of this application were packaged inside COM Servers and most of them are ActiveX Dlls. These Dlls were created by using VB6. The C# application and these Dlls are communicating by using Com and Net Interoperability. ADO was used in order these Dlls to manipulate the XSLT Rules and create some other output files (such as the customized views for the products) and also to do some clean up. ADO.NET was used in order the C# application to communicate with the SQL Server. Data again were stored in the SQL Server 2000. FSO (File System Object Model) was also employed in order to do some File Manipulation, for some intermediate text files created during the process of the customization.

Using Com Objects from .NET

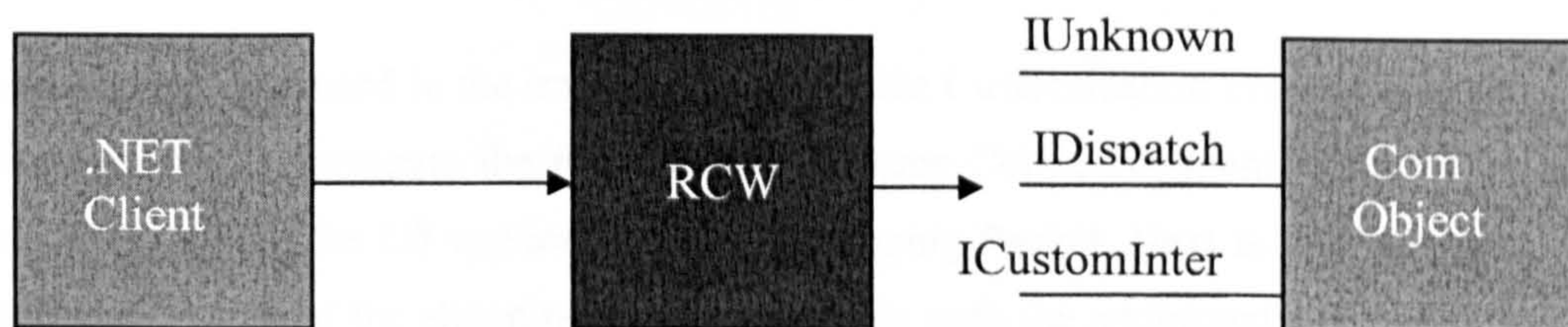


Figure 17: Client Access through RCW

In order to develop the RCW (Runtime Callable Wrapper) we can employ two different methods. One is to use the Visual Studio .NET and right-click at the Reference Section of the Project and select Add Reference from the Context Menu. A

dialog box offers a choice of all the registered COM type libraries it finds on the system, and also allows you to select unregistered type libraries by browsing to them (see **Figure 18**). We select the COM object for which we want to generate the RCW, and Visual Studio .NET will create it and add it to the project.

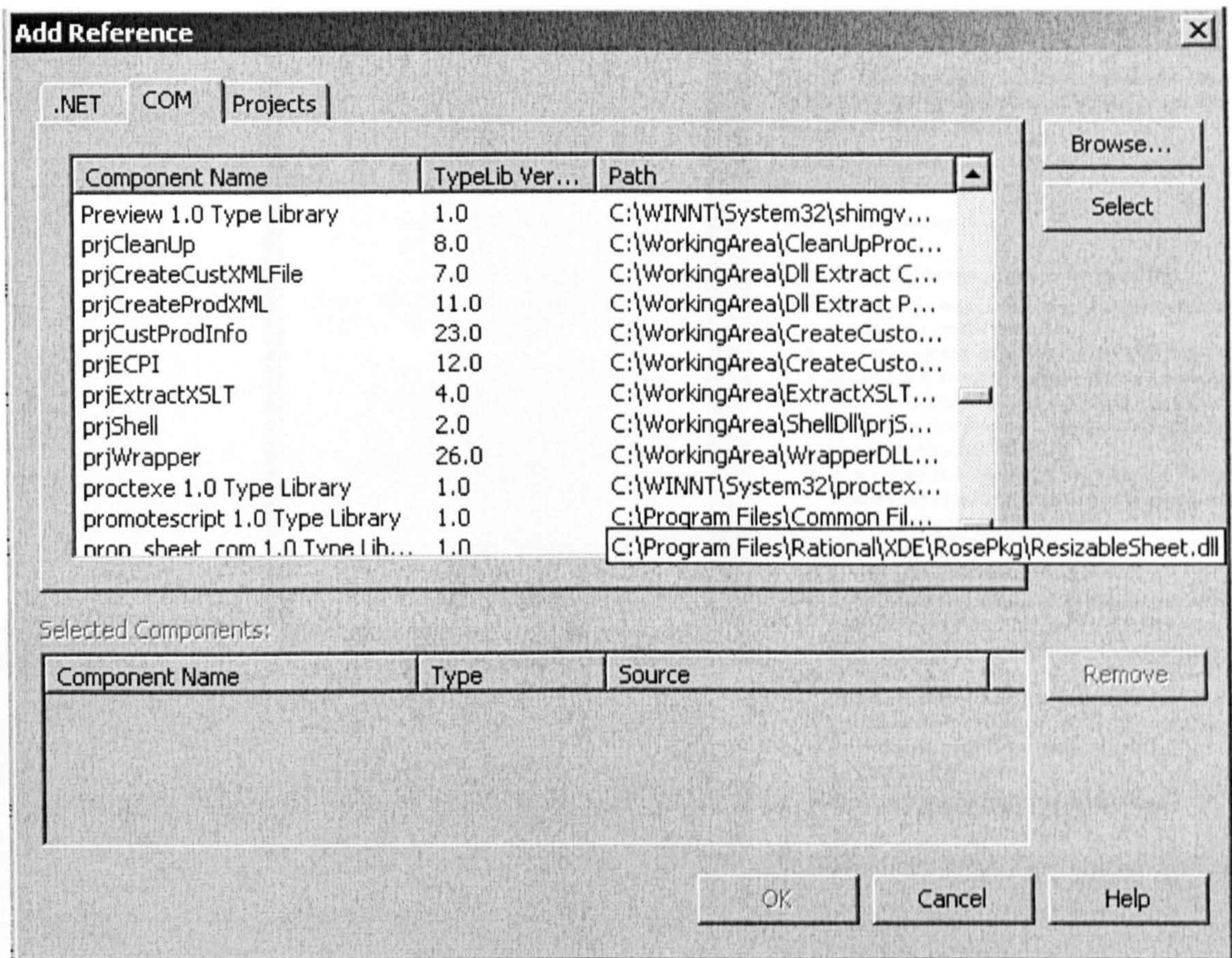


Figure 18: Generating a RCW.

This technique was used in the implementation of the Customization Process in order to combine and incorporate the functionality of some Com components (created by using VB6) within the C# application of the Shopping Basket. Next is the overview of the architecture of the shopping basket alongside with the additional functionality provided by the COM objects (Helper Classes I called them).

4.4.1 System Description

XSLT Rules Creator Application Description

Main Interface

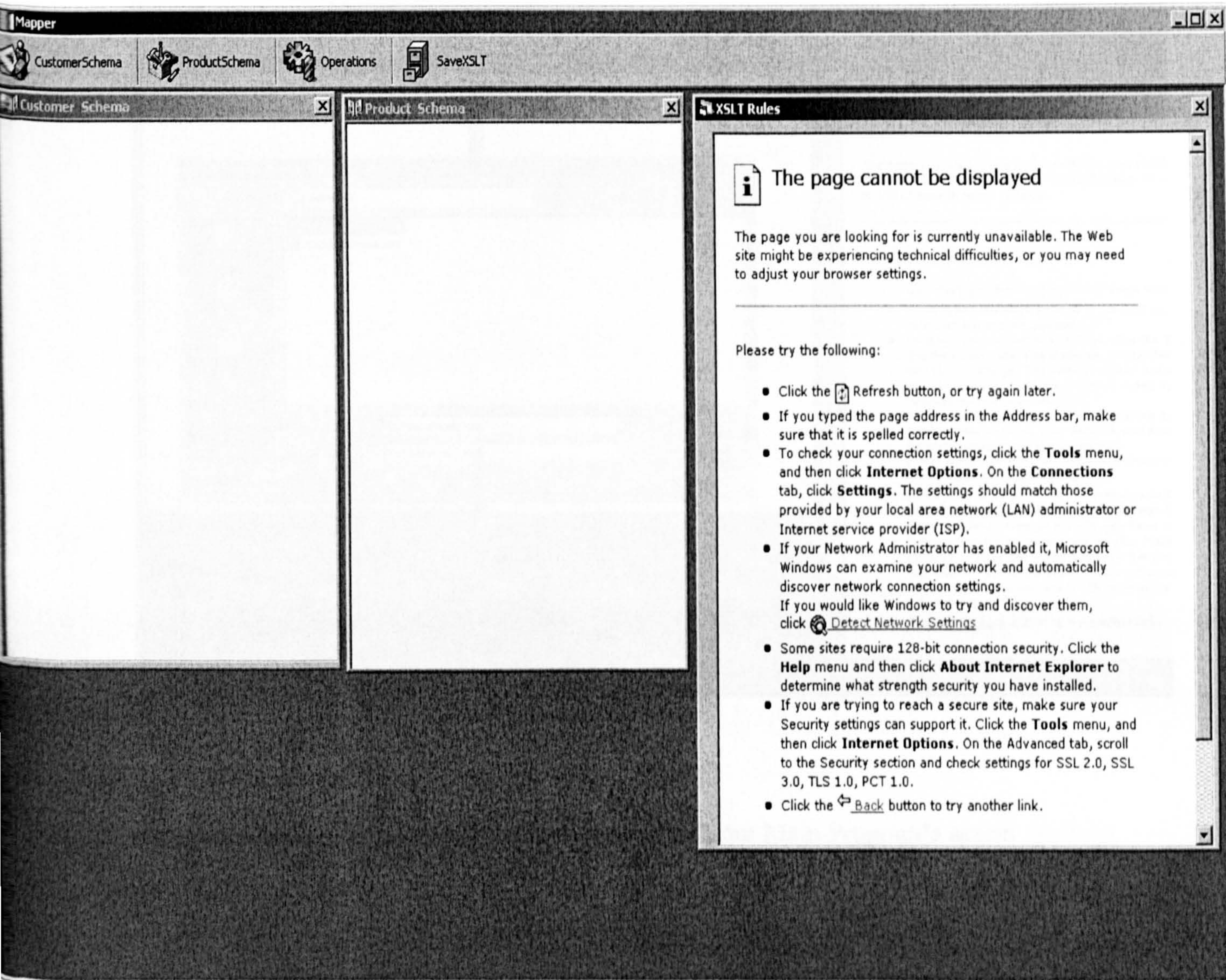


Figure 19: XSLT Rules Creator Application (Main Interface). Is used to carry out the XSLT Rules Creation and the storage of them in the database.

This is the main interface, which is displayed whenever the user starts the program. All the available functions are being displayed in the Action Bar at the top of the screen. The user is provided with the ability to load a Customer Schema with the *CustomerSchema* action. Also to load a Product Schema with the *ProductSchema* action. This main interface was created as a standard Exe program. The action *Operations* is being used in order to perform some associations between the Customer Schema and the Product schema elements. This is actually the action which is being used in order to create the XSLT rules which are in a way associate the Customer Segment with a specific Product Category. The action *SaveXSLT* saves the produced XSLT Rules in the database.

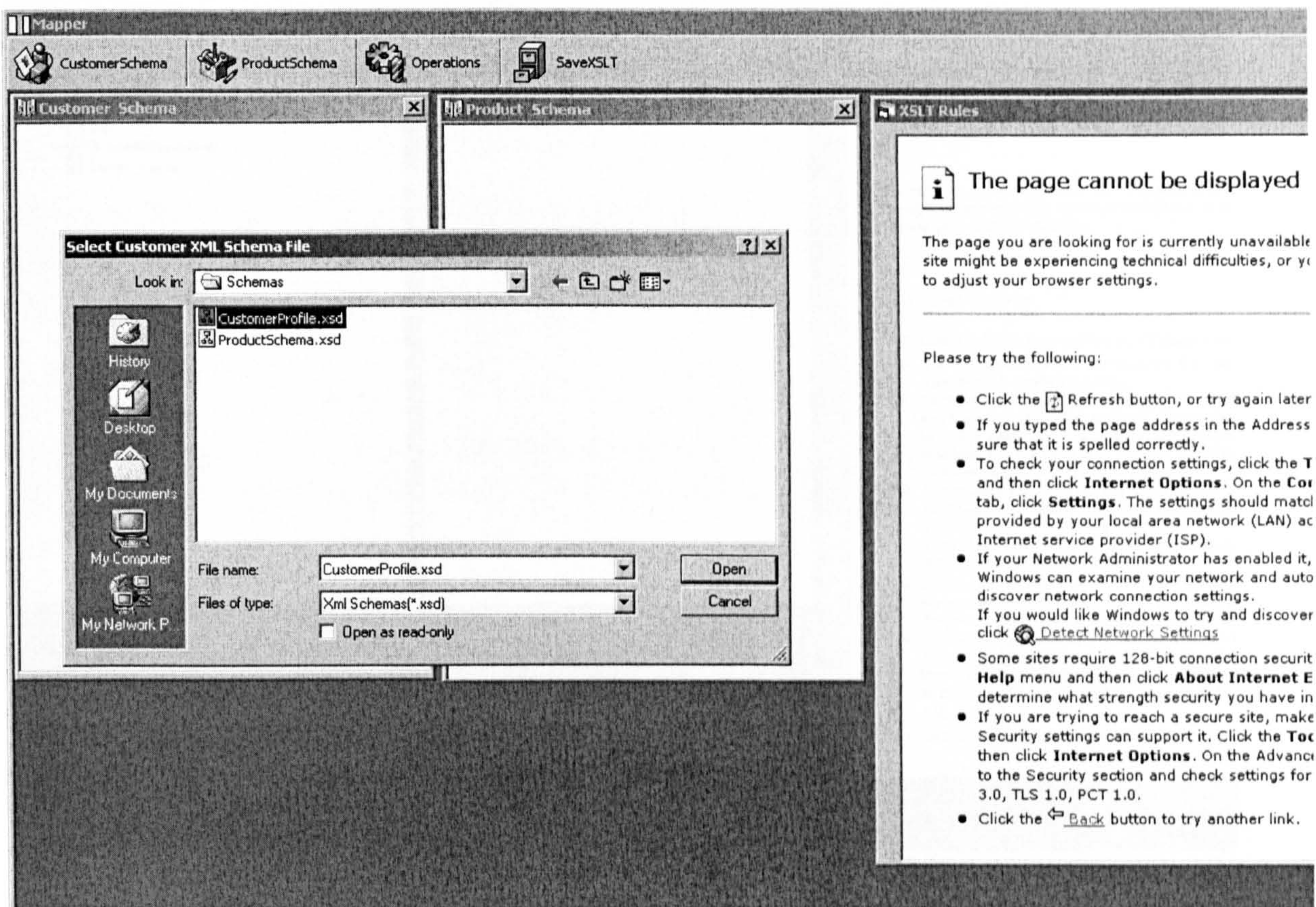


Figure 20: Select Customer XML Schema File

Whenever the user selects *CustomerSchema* action from Main Program's action bar is being provided with the interface shown in Figure 20. This is a Dialog Box used in order to select a Customer Schema and by selecting the *Open* command to load it inside a window form with caption *Customer Schema*. The Customer Schema is being represented by using a Tree View Control. Each element of the Customer Schema is represented as a node in the tree. Figure 21 shows the Customer Schema in a Tree Format.

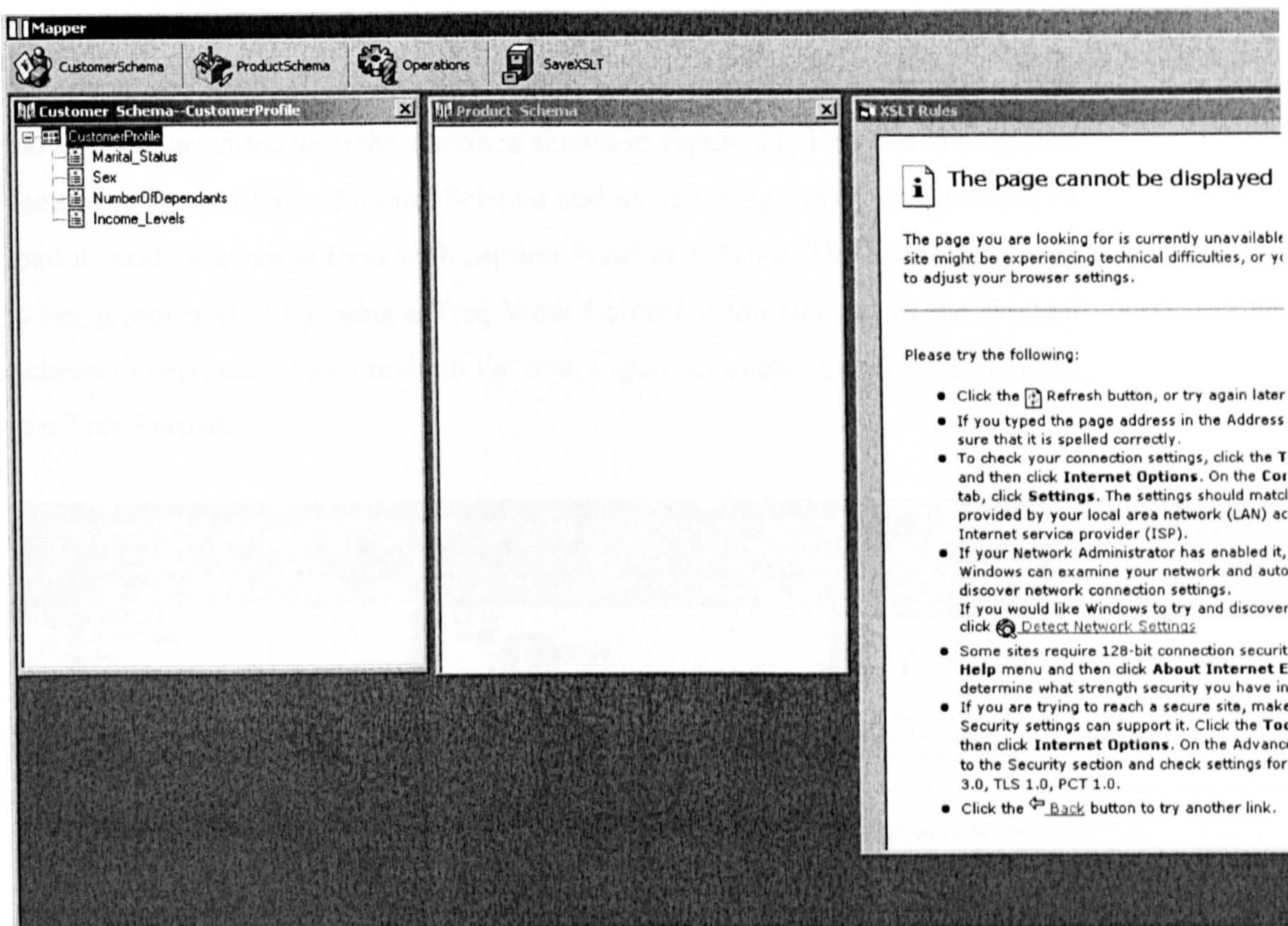


Figure 21: The Customer Schema populates a Tree View Control and each element of the Customer Schema is being represented as a node in the tree.

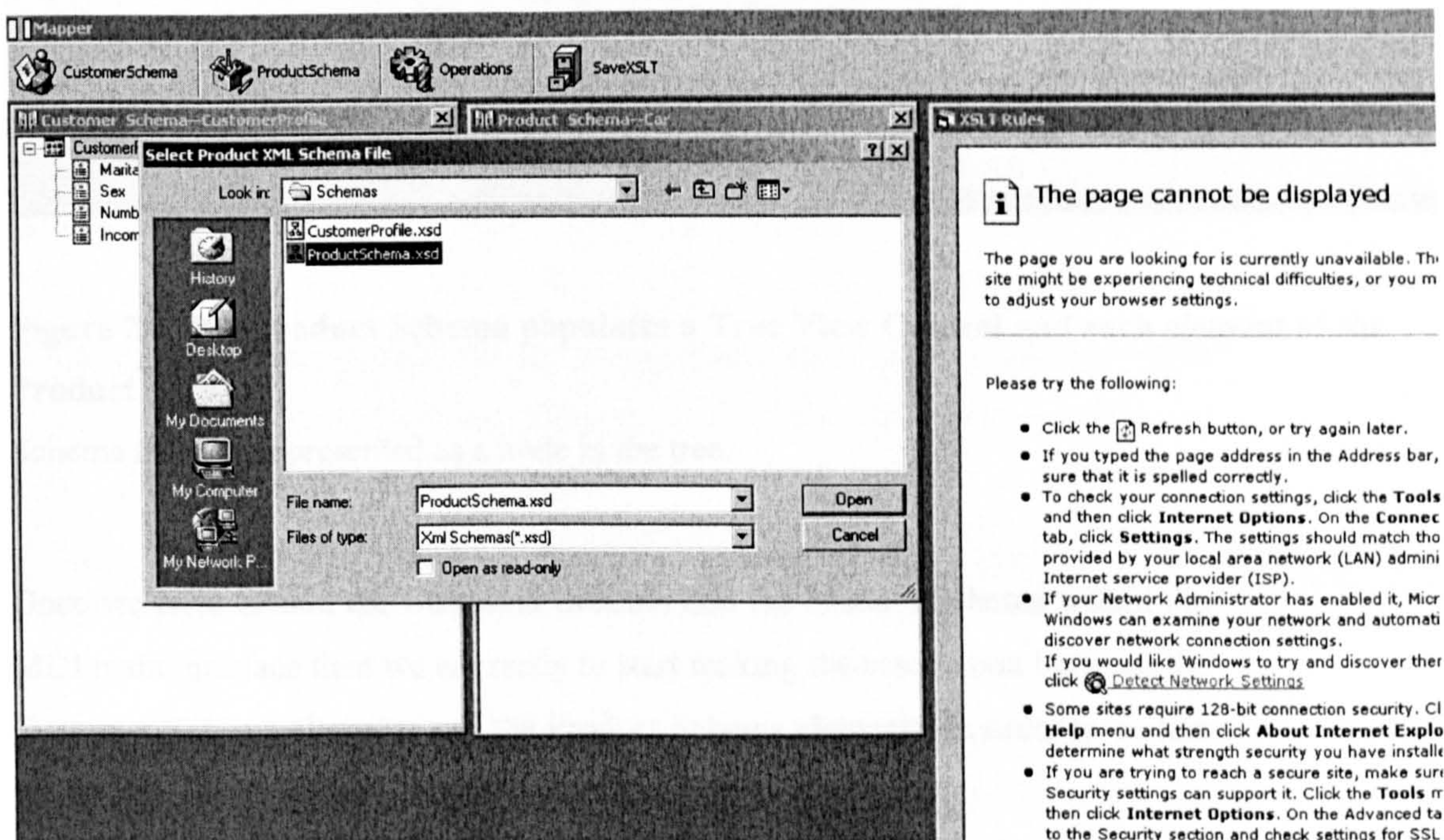


Figure 22: Select Product XML Schema File

Whenever the user selects *ProductSchema* action from Main Program's action bar is being provided with the interface shown in Figure 22. This is a Dialog Box used in order to select a Product Schema and by selecting the *Open* command to load it inside a window form with caption *Product Schema*. The Product Schema is being represented by using a Tree View Control. Each element of the Product Schema is represented as a node in the tree. Figure 23 shows the Product Schema in a Tree Format.

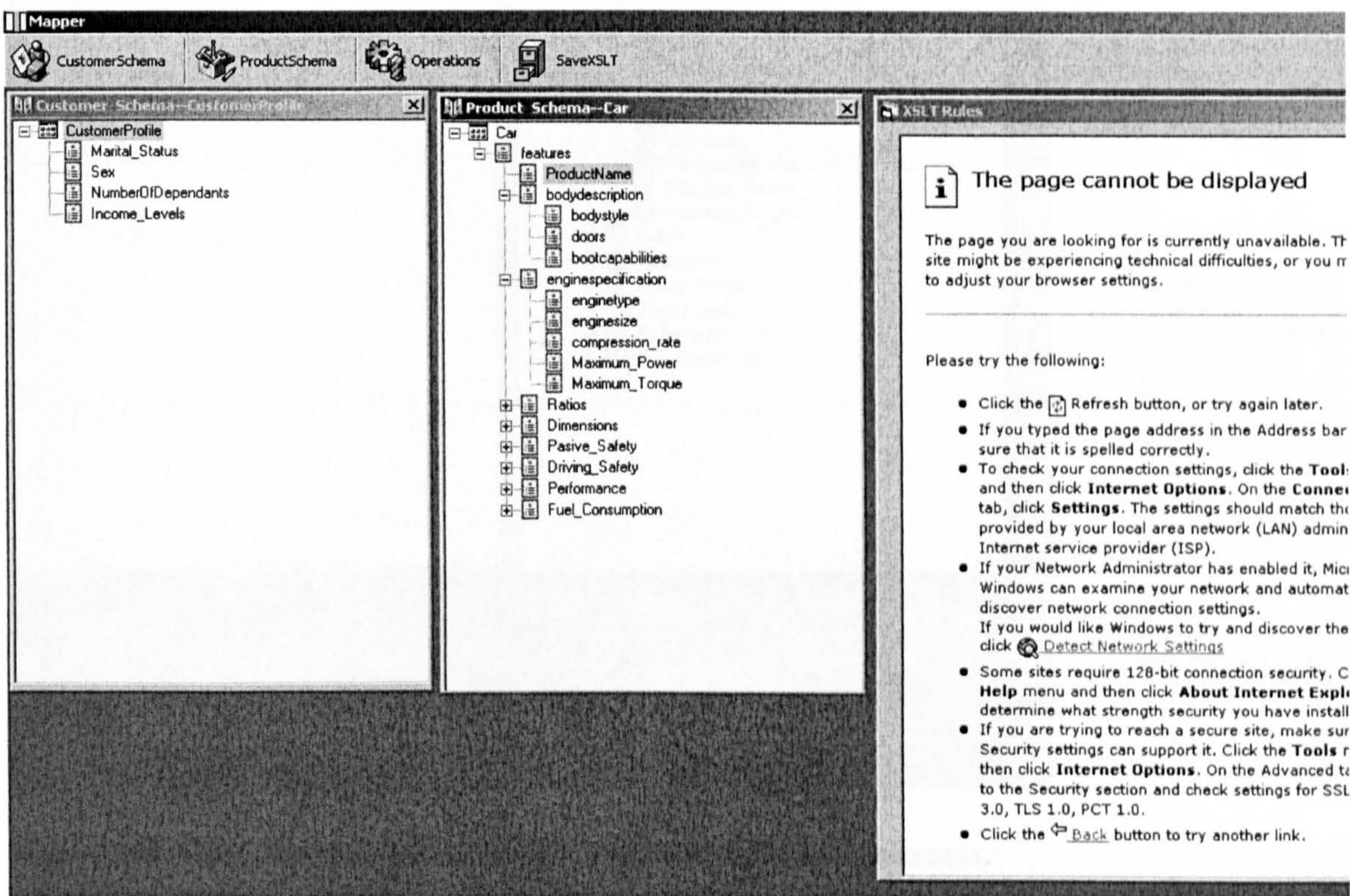


Figure 23: The Product Schema populates a Tree View Control and each element of the Product

Schema is being represented as a node in the tree.

Once we have loaded the Customer Schema and the Product Schema within the MDI main interface then we are ready to start making the association between the Customer Schema elements and the Product Schema elements. In order to do this we have to perform the following actions:

1. Right click on the MDI Main Window. A Context Window Appears.
Select a XSLT Statement - Construct (The ones that have been

implemented here for demonstration purposes are *If Then Else* and *Nested If* Constructs). Also the *Choose* Construct can be implemented. The Context Menu which appears when we Right Click on the Main Interface is shown in figure 24.

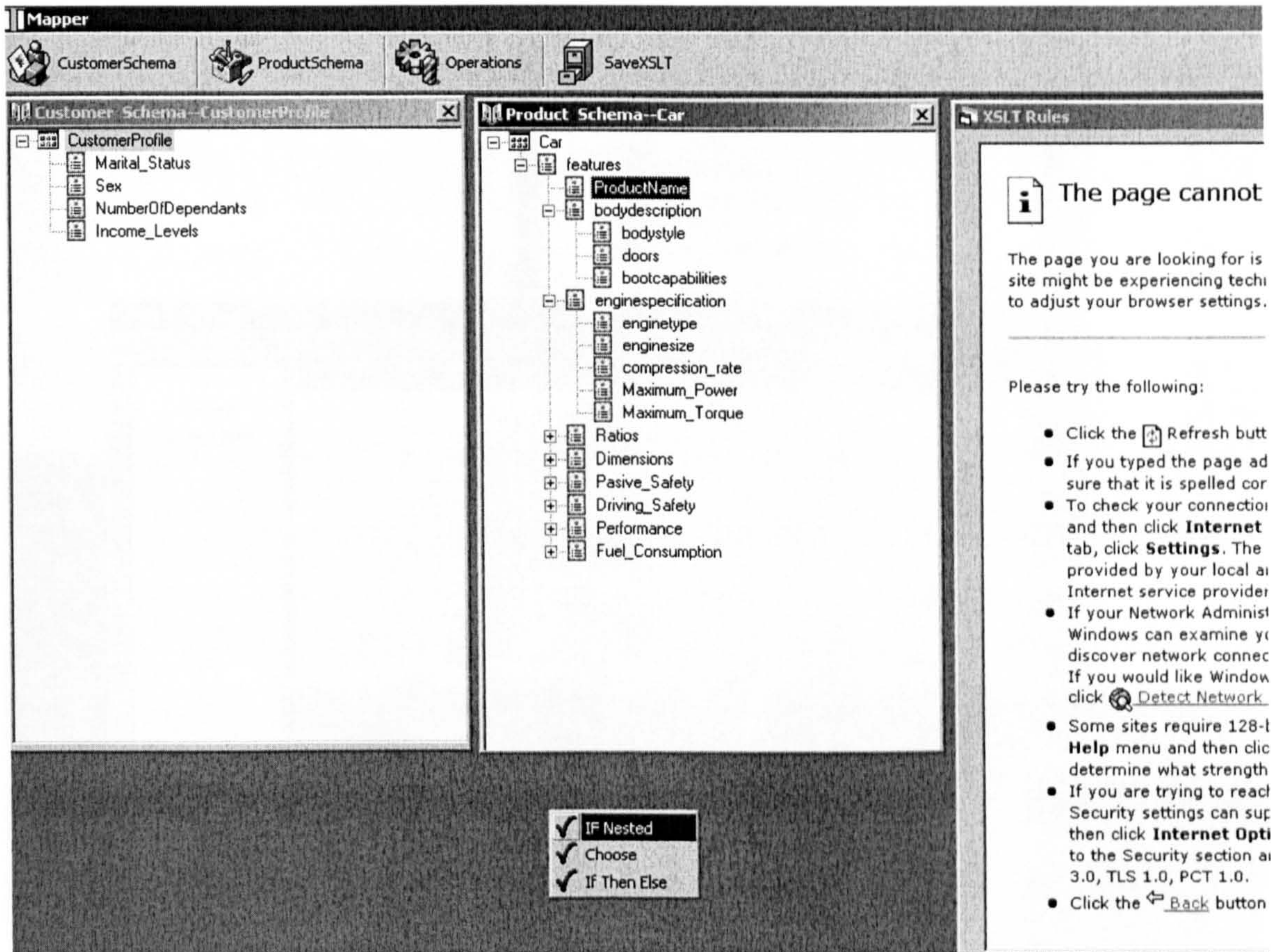


Figure 24: The Context Menu with the available XSLT Constructs.

- Once we have selected the XSLT Construct then we select the *Operations* Action from the Action Bar. Then a new Window Form is activated. It works as a workbench and is being used to create a fragment of the XSLT Rules. Its caption is Operations. Contains two Data Grid controls used to accommodate the Customer Schema Elements and the Product Schema elements. These elements are dragged over from the Tree Controls into these Data Grid Controls. One Data Grid is called CustomerItems and the other ProductItems. There is also another control within the *Operations* form which is an *Action Bar*. This Action Bar titled *Conditional* contains 3 XSLT constructs. *If*, *Choose* and *If Then Else*. The *Operations* Windows Form is shown in Figure 25.

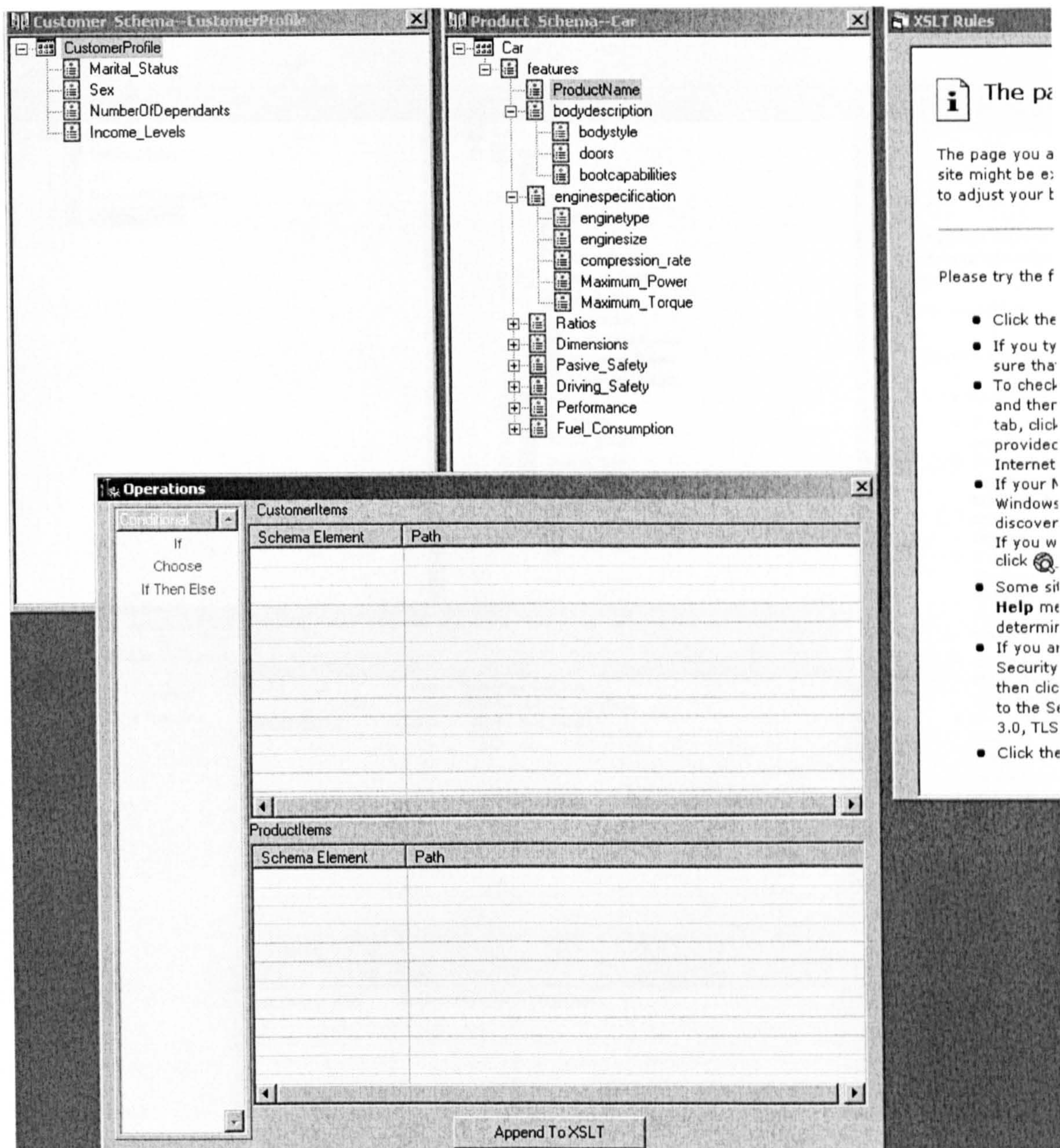


Figure 25: The Operations Windows Form.

3. The next step is to start performing the Drag And Drop Operation between the Customer Schema and the Product Schema Tree View Controls and the *CustomerItems* and *ProductItems* Data Grid Controls. The populated *CustomerItems* and *ProductItems* Data Grid Controls are shown in Figure 26.

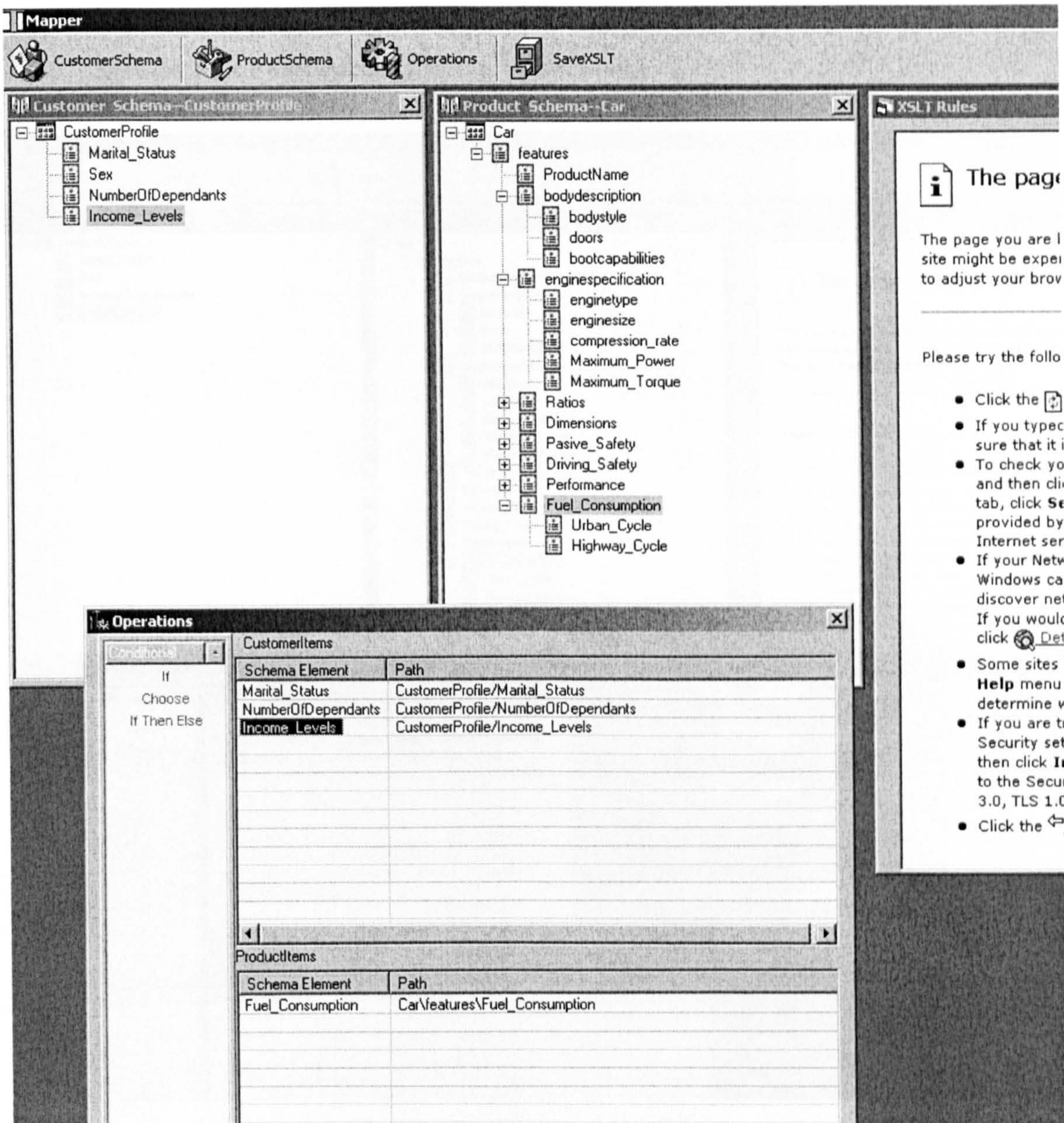


Figure 26: The Operations Form Populated with Customer and Product Schema elements.

- Once we have dragged the Customer Schema and the Product Schema elements in the Data Grid Controls then we select an operation from the *Conditional* Action Bar. Whenever a user selects an operation then a new window form is being displayed which allows the user to create a fragment of a XSLT Rule. He/she does so by creating some conditional statements related to some Customer Schema Elements and then depending on the validity of these statements a specific element from the Product Category is selected. The form which facilitates the creation of a XSLT Rule is shown in Figure 27. In this particular one I have selected to create an If (Nested) statement. The Form which facilitates the creation of

this XSLT statement (which is actually a fragment of the entire XSLT Rules set) has a caption with the title *If Condition*.

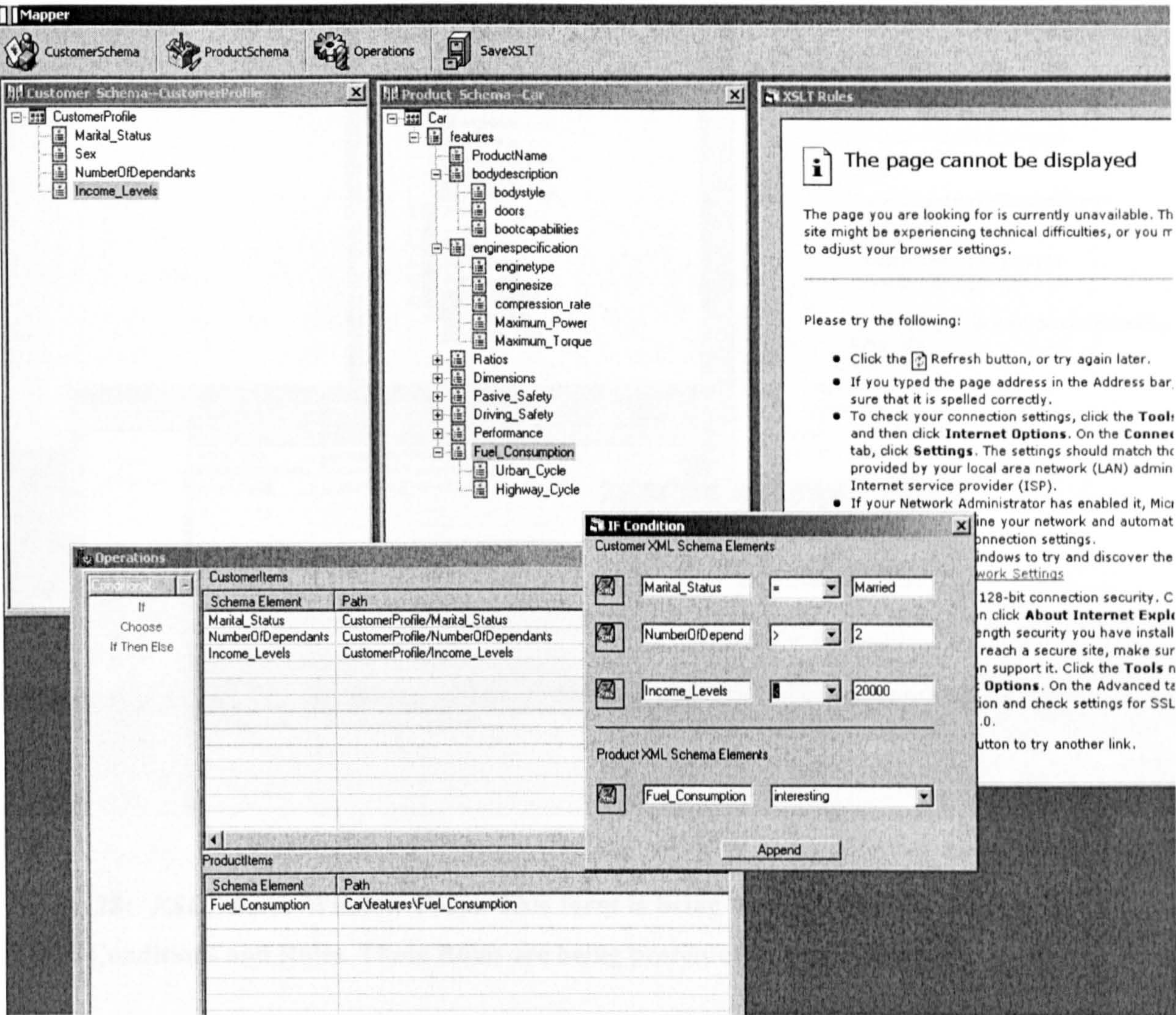


Figure 27: *If Condition* Window Form.

- Once we have finished with the creation of the XSLT Rule then we press the Append Command Button and then the XSLT Rule is being attached to the Main body of the XSLT Rules which is being displayed in a different window which has a caption *XSLT Rules*. The window form with caption *XSLT Rules* which accommodates the XSLT Rules – Conditions is shown in Figure 28.

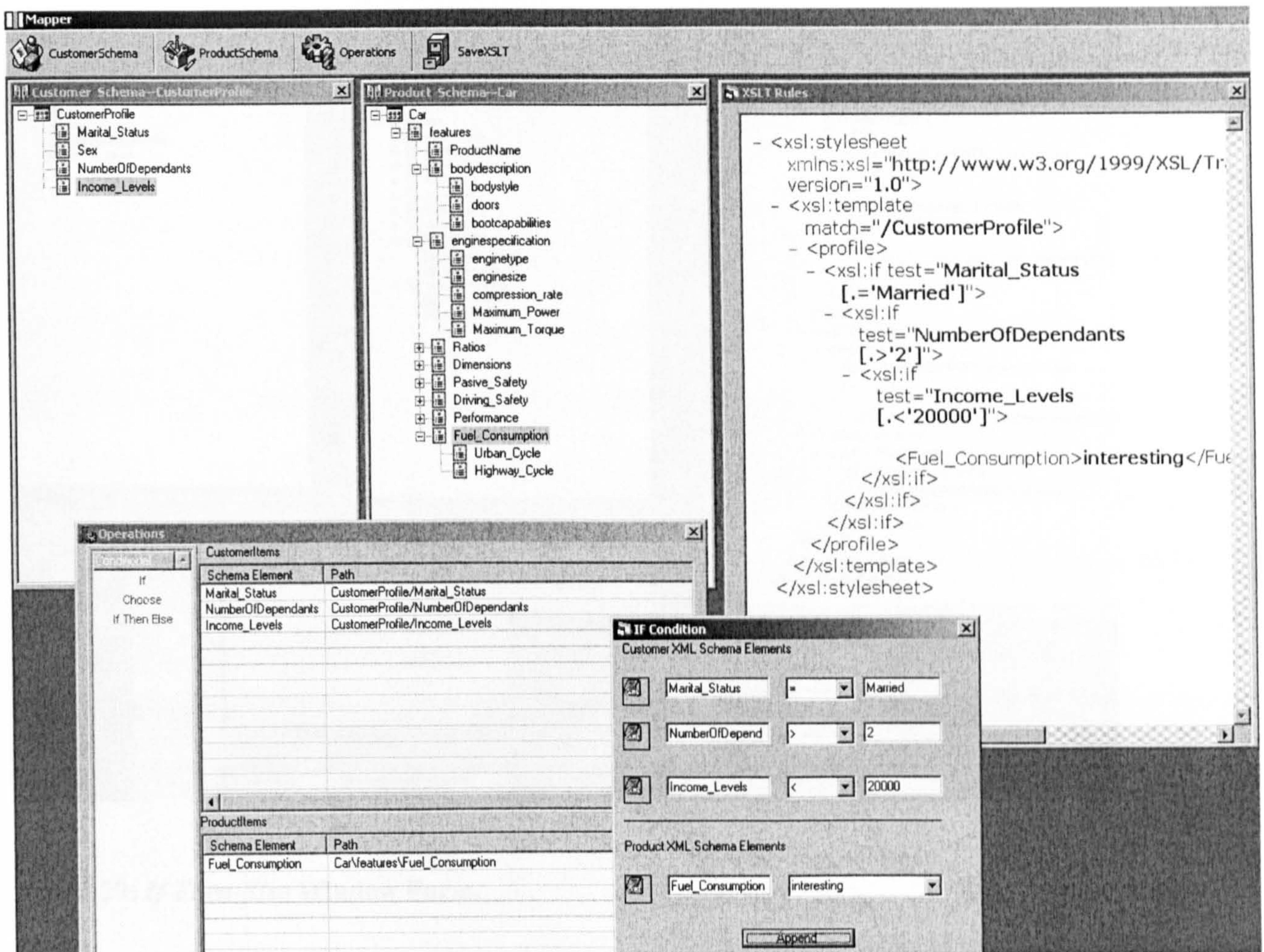


Figure 28: XSLT Rules Window Form. This form is being used in order to display the XSLT Conditions and Rules. These Rules are being presented in a tree structure.

- We can repeat again the steps from 2 to 5 and select another kind of XSLT Construct. In this case I have selected the *If Then Else* Construct. Here we associate one element from the Customer Profile with two different elements from the Product Category Schema. Once we have produced the XSLT Rule then we select the Append command Button and the XSLT Rule is attached to the Main Body of the XSLT Rules which resides within the *XSLT Rules Window Form*. The result of this process can be seen in the Figure 29.

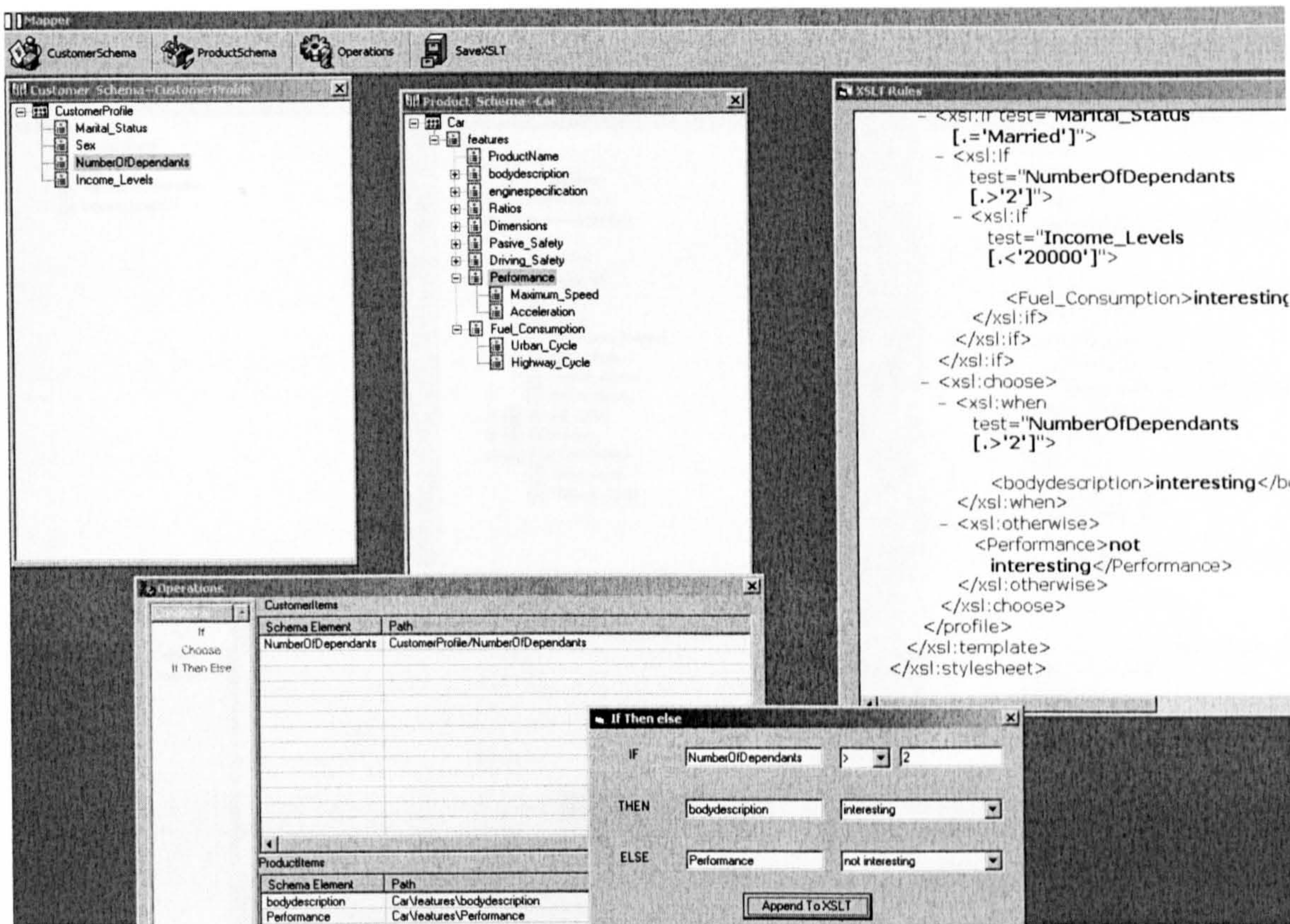


Figure 29: If Then Else Window Form.

- For the sake of demonstration I'll create another XSLT Rule by using the If Then Else Construct. The Resulting Screen Shot is shown in Figure 30.

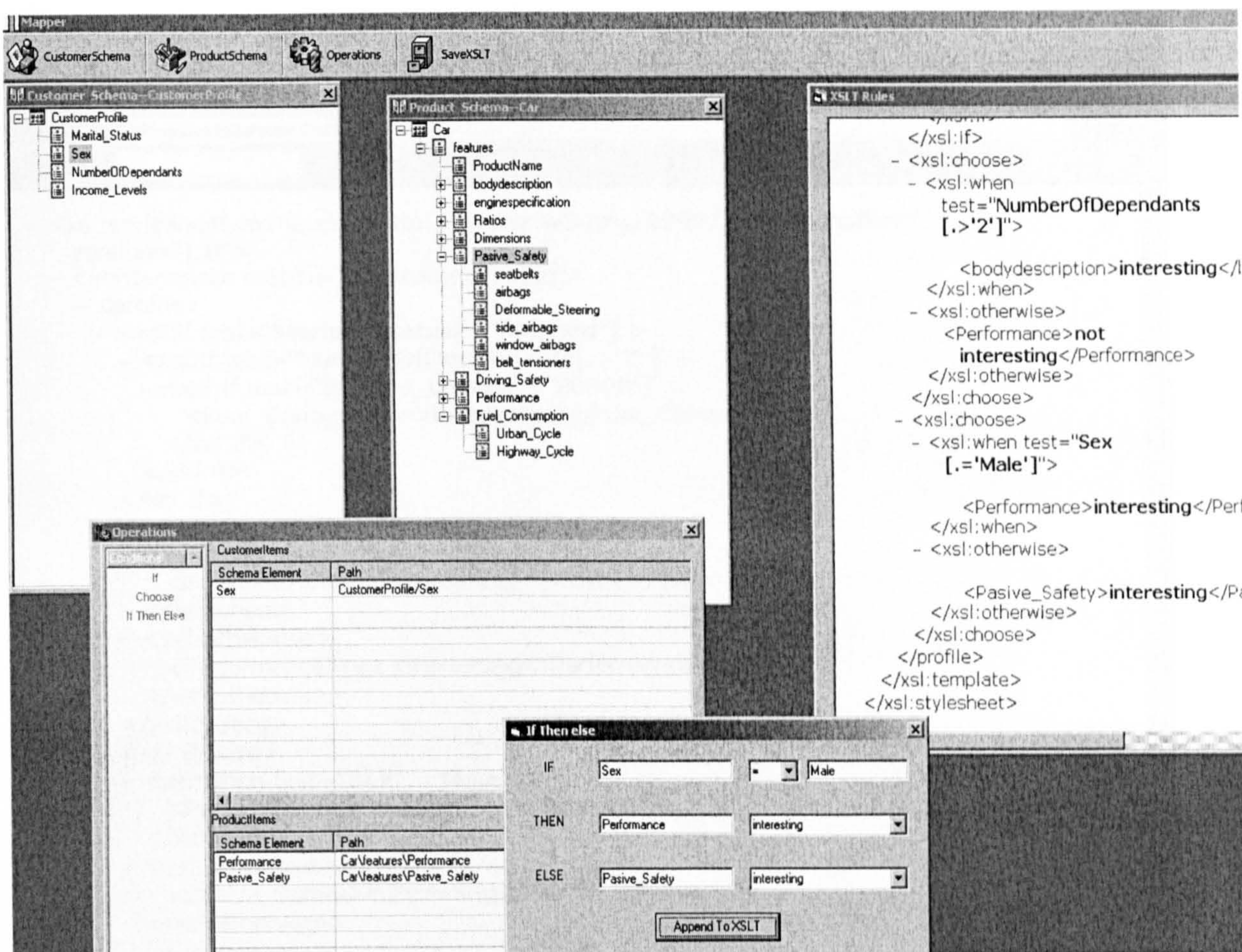


Figure 30: If Then Else Window Form. Another If Then Else Construct was used for the sake of demonstration.

- When we have finished with the creation of the XSLT Rules then the next step is to save them. If we select *SaveXSLT* action from the Main Window's Action Bar then the XSLT Rules are stored automatically in the database (SQL Server 2000). The XSLT Rules are stored as a string of ASCII characters. The set of the created XSLT Rules can be seen in Figure 31.

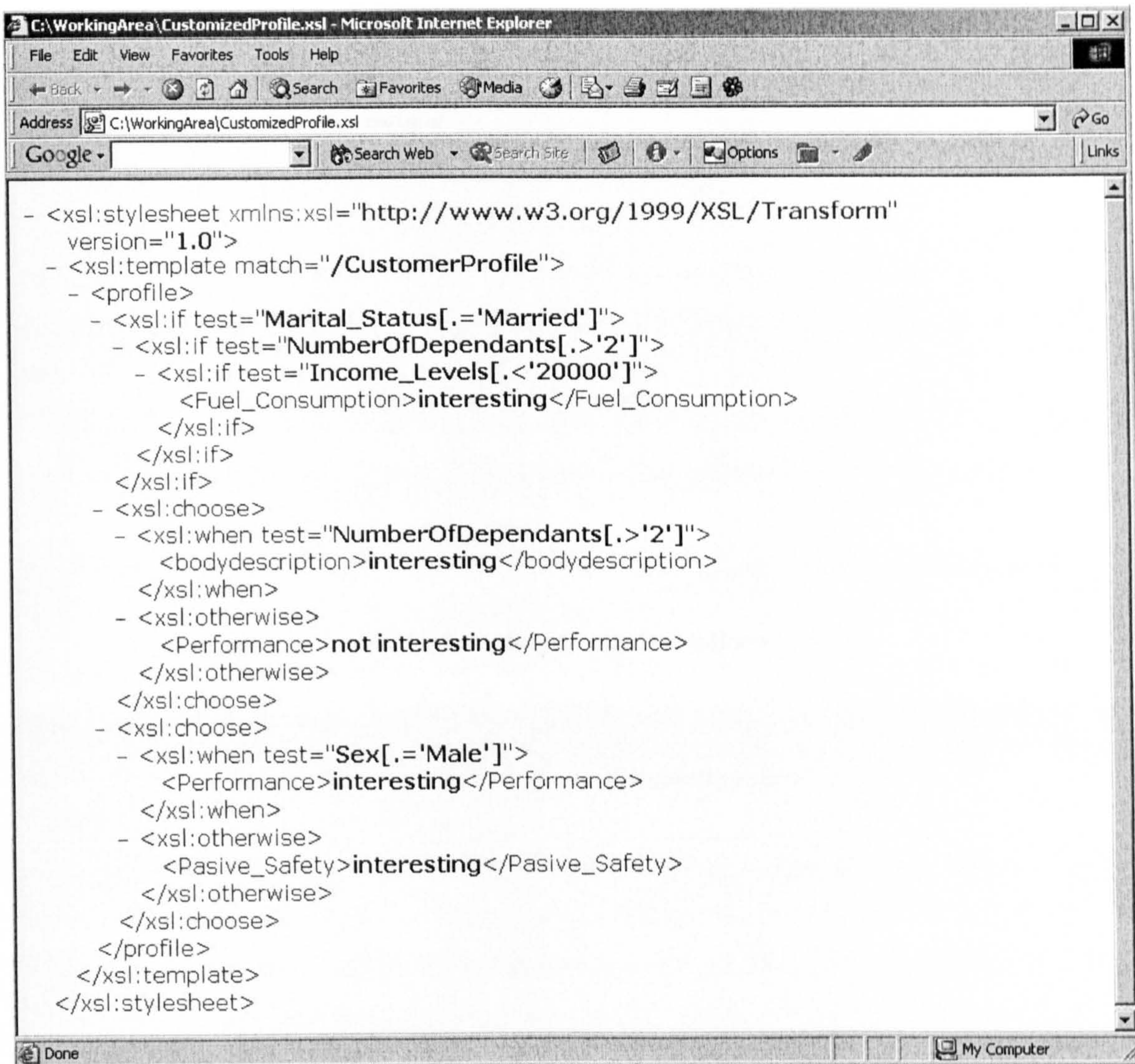


Figure 31: Set of XSLT Rules.

These rules as can be seen create a relationship between the Elements of a Customer Segment and a Product Category. This relationship is in Schema Level. When a schema is instantiated, e.g. when we create a XML file which complies with this schema then we test these rules against the XML files of the Product Category and the Customer Segment. Now that we have finished with the one set of the XSLT Rules we will create another one in order to do the testing of the application and see how the Customization process works for different Customers. The steps 1 to 8 have been performed once again and the second set of XSLT Rules looks can be seen in Figure 32. This set of XSLT Rules is to be associated with a second customer with ID Customer2.

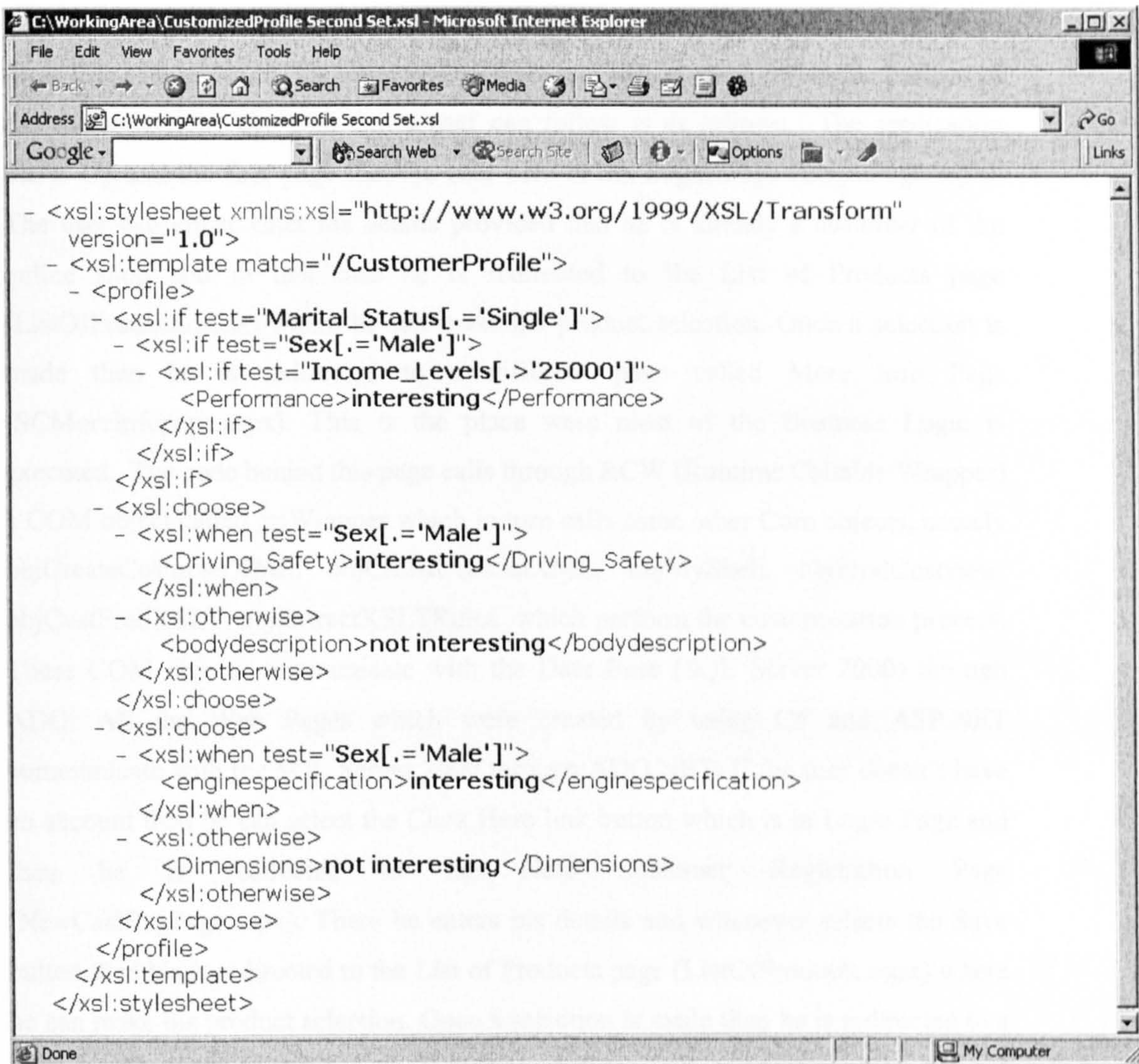


Figure 32: Second Set Of XSLT Rules.

Shopping Basket – Testing of Customization Process

This is the second part of the implementation. The Shopping Basket is a three-tier application. The Main interface is a Web Based Interface (Internet Explorer). It was created by using C#, ASP.NET. Some other constituent parts of this application were packaged inside COM Servers and most of them are ActiveX Dlls. These Dlls were created by using VB6. The C# application and these Dlls are communicating by using Com and Net Interoperability. ADO was used in order these Dlls to manipulate the XSLT Rules and create some other output files (such as the customized views for the products) and also to do some clean up. ADO.NET was used in order the C# application to communicate with the SQL Server. Data again were stored in the SQL

Server 2000. FSO (File System Object Model) was also employed in order to do some File Manipulation, for some intermediate text files created during the process of the customization. The path that a user can follow is as follows. The application Starts Up and the first page that the user sees is the Login Page (LoginPage.aspx). The user can either enter his details provided that he is already a customer of the online shop and in this case he is redirected to the List of Products page (ListOfProducts.aspx) where he can make his product selection. Once a selection is made then he is redirected to a different page called More Info Page (SCMoreInfopage.aspx). This is the place where most of the Business Logic is executed. The code behind this page calls through RCW (Runtime Callable Wrapper) a COM object called prjWrapper which in turn calls some other Com objects, namely objCreateCustomerXML, objCreateProductXML, objMyShell, objProdCustview, objCustProdXSLT, objExtractXSLTRules which perform the customization process. These COM objects communicate with the Data Base (SQL Server 2000) through ADO. All the Web Pages which were created by using C# and ASP.NET communicate with the SQL Server 2000 through ADO.NET. If the user doesn't have an account then he can select the Click Here link button which is in Login Page and then he is redirected to the New Customer Registration Page (NewCastRegPage.aspx). There he enters his details and whenever selects the Save button then he is redirected to the List of Products page (ListOfProducts.aspx) where he can make his product selection. Once a selection is made then he is redirected to a different page called More Info Page (SCMoreInfopage.aspx). The overall application architecture can be seen in Figure 33.

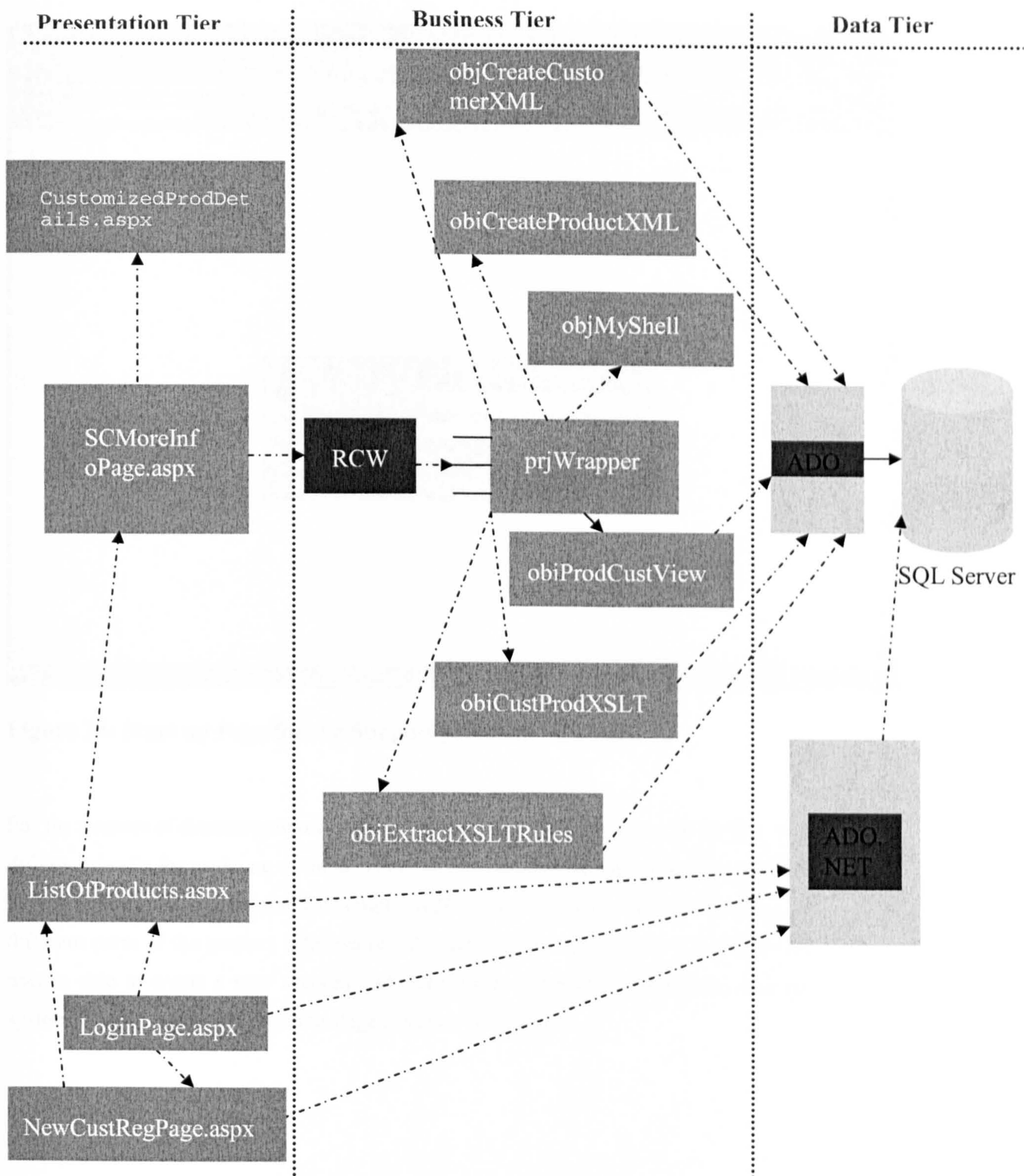


Figure 33: Application Architecture.

This is the first page, which is displayed whenever the user starts the program. The user is provided with the ability to either Login to the System if he/she is already has an account or creating an account by clicking at the Link Button called Click Here. The start up page is shown in figure 34.

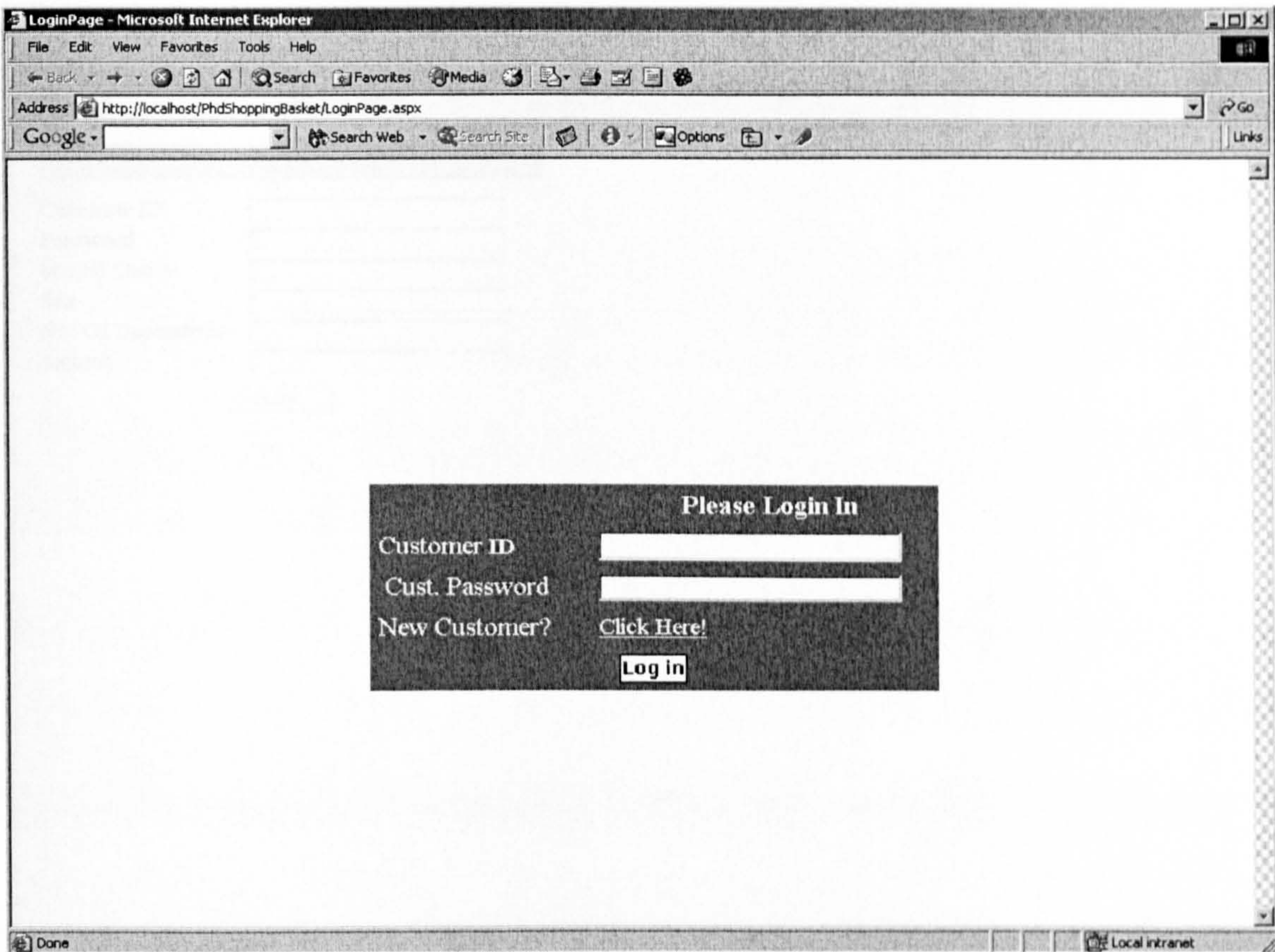


Figure 34: Start up Page for the Shopping Basket.

For the purpose of demonstration two different customer accounts are to be created with different details for each one of them. Then we will see that depending on the different XSLT Rules related to these Customer profiles the customers will be looking at different parts of the product information. By selecting the *Click Here* link button the user is able to create a new account and store some of his/hers information into the system. The Customer Registration Page is shown in Figure 35.

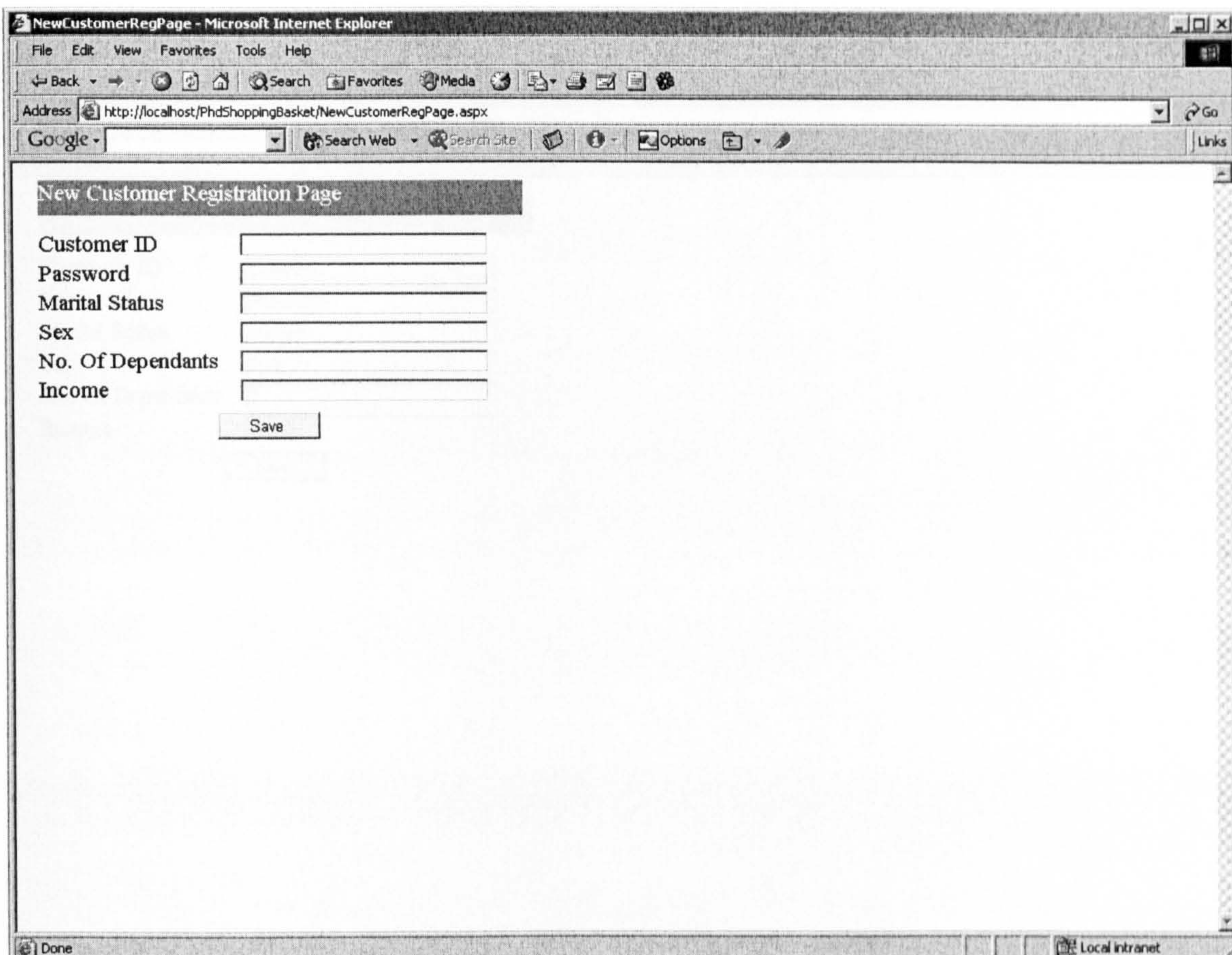


Figure 35: New Customer Registration Page

We fill in the required information. All the entered information is shown in Figure 36. The data entered for the first customer profile they are selected in purpose in order to comply with the first set of XSLT Rules which can be seen at Figure 31. When the user selects **Save** then he/she is transferred to the product list (This product list displays some car models). The Product List is shown in Figure 37.

NewCustomerRegPage - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Options Links

Address http://localhost/PhdShoppingBasket/NewCustomerRegPage.aspx Go

Google Search Web Search Site Options Links

New Customer Registration Page

Customer ID	<input type="text" value="Customer1"/>
Password	<input type="text" value="Password1"/>
Marital Status	<input type="text" value="Married"/>
Sex	<input type="text" value="Male"/>
No. Of Dependants	<input type="text" value="3"/>
Income	<input type="text" value="18000"/>

Done Local intranet

Figure 36: First Customer's Profile.

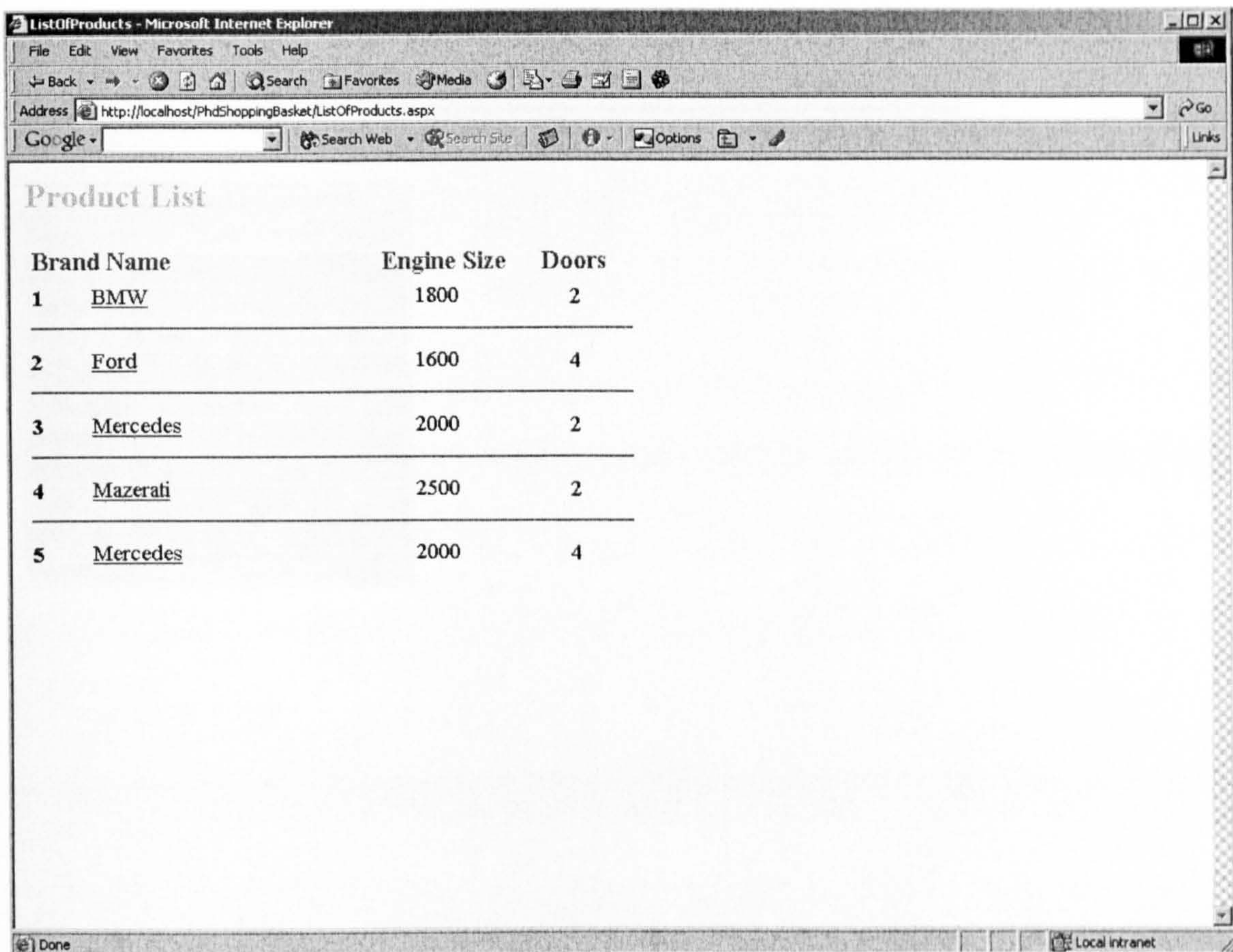


Figure 37: Product List.

If the user makes a selection then is represented with a customized view for the specific product. If a Customer makes a selection e.g. No 5 from the Product List, which is a Mercedes with a 4 doors and an engine size of 2000 then is represented with the Product Information shown in Figure 38.

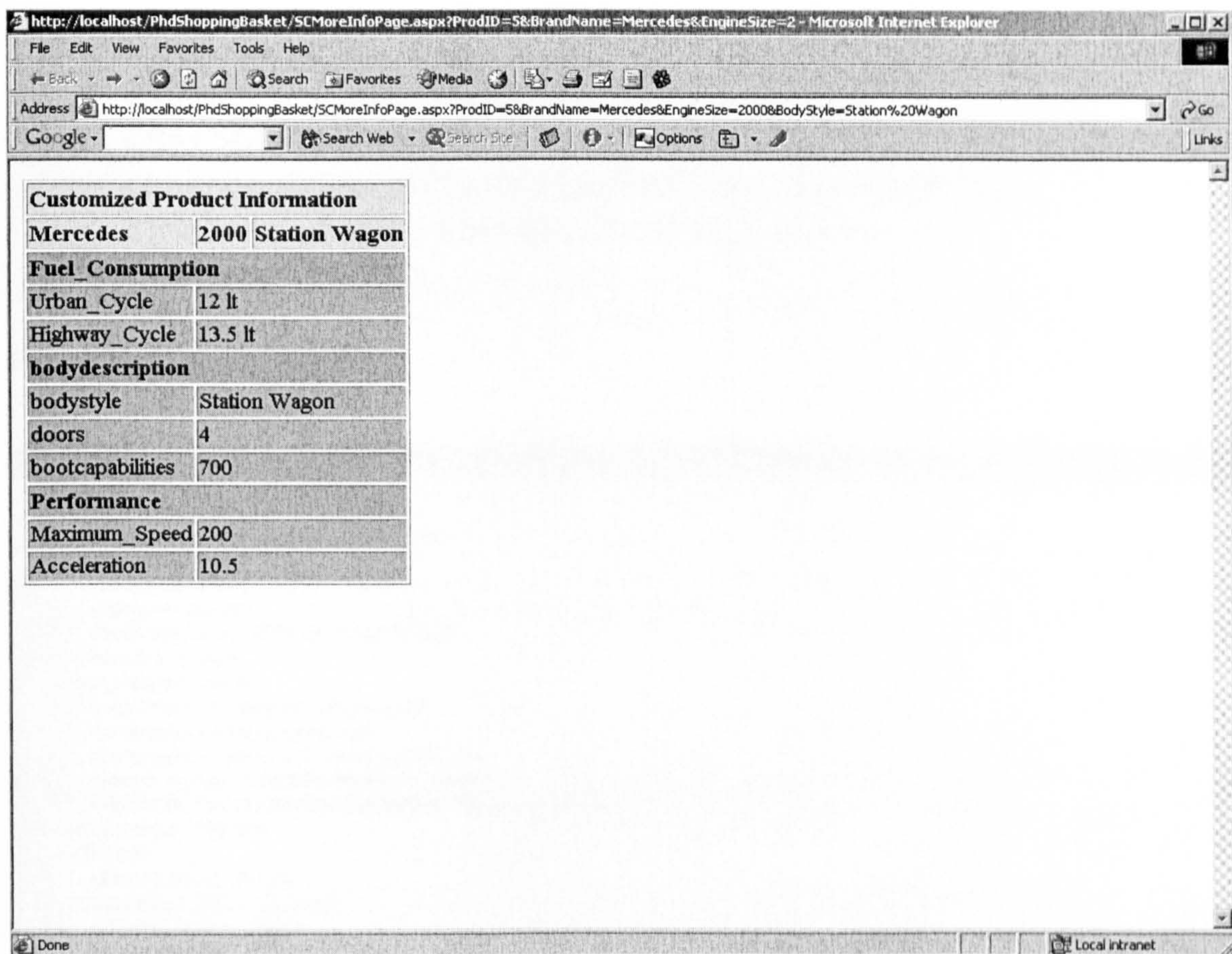


Figure 38: Customized view for Customer1.

Inner working – How does this work?

We have to concentrate on three different business documents. The Customer Profile, the Product itself (Figure 40) and the set of XSLT rules which actually creates a relationship between a specific customer category and a product category. The Customer Profile for the first customer is shown In Figure 39:


```

- <CustomerProfile>
  <Marital_Status>Married</Marital_Status>
  <Sex>Male</Sex>
  <NumberOfDependants>3</NumberOfDependants>
  <Income_Levels>18000</Income_Levels>
</CustomerProfile>

```

Figure 39: Customer Profile



```

- <Car>
- <features>
  <ProductName>Mercedes</ProductName>
- <bodydescription>
  <bodystyle>Station Wagon</bodystyle>
  <doors>4</doors>
  <bootcapabilities>700</bootcapabilities>
</bodydescription>
- <enginespecification>
  <enginetype>6 Cylinder</enginetype>
  <enginesize>2000</enginesize>
  <compression_rate>7.0:1</compression_rate>
  <Maximum_Power>140HP</Maximum_Power>
  <Maximum_Torque>2000Nm@4000rpm</Maximum_Torque>
</enginespecification>
- <Ratios>
  <First>2.876:1</First>
  <Second>1.200:1</Second>
  <Third>1.300:1</Third>
  <Fourth>0.500:1</Fourth>
  <Fifth>0.654:1</Fifth>
</Ratios>
- <Dimensions>
  <Overall_Length>4000.5</Overall_Length>
  <Overall_Height>1700</Overall_Height>
</Dimensions>
- <Pasive_Safety>
  <seatbelts>4</seatbelts>
  <airbags>4</airbags>
  <Deformable_Steering>Yes</Deformable_Steering>
  <side_airbags>Yes</side_airbags>
  <window_airbags>yes</window_airbags>
  <belt_tensioners>yes</belt_tensioners>
</Pasive_Safety>
- <Driving_Safety>
  <ABS>yes</ABS>
  <Brake_Assist>yes</Brake_Assist>
  <ESP>No</ESP>
  <ASR>No</ASR>
</Driving_Safety>
- <Performance>
  <Maximum_Speed>200</Maximum_Speed>
  <Acceleration>10.5</Acceleration>
</Performance>
- <Fuel_Consumption>
  <Urban_Cycle>12 lt</Urban_Cycle>
  <Highway_Cycle>13.5 lt</Highway_Cycle>
</Fuel_Consumption>
</features>
</Car>

```

Figure 40: Specific Product XML File.

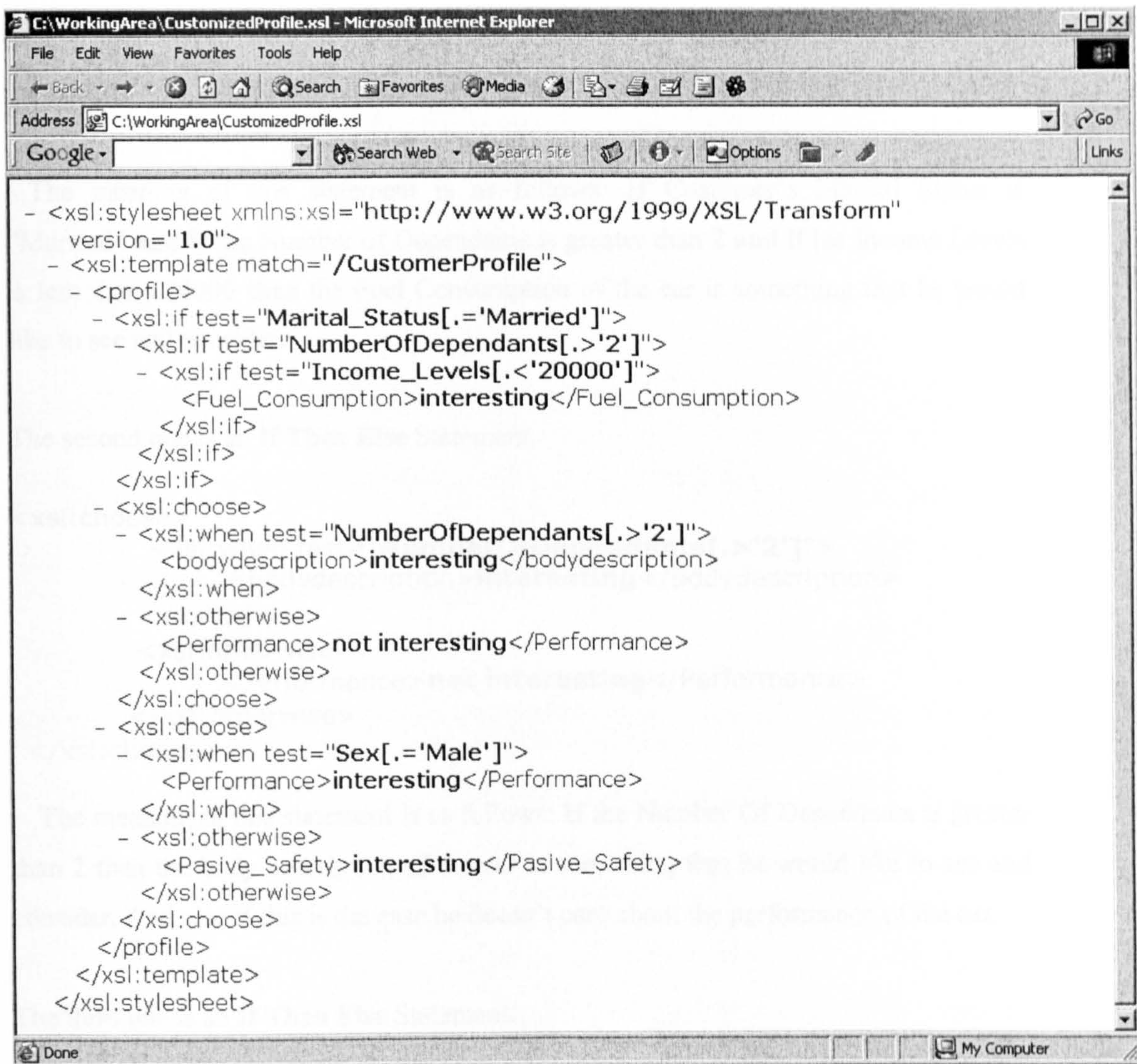


Figure 41: XSLT Rules for Customer.

The third business document is the XSLT Rules for the Customer Category in which the first customer belongs (Figure 41). Is obvious that this business document creates an association between a specific product category, namely cars, and a specific customer category in which our first customer belongs. This set of XSLT Rules contains actually three different conditional statements. The first one is a nested if:

```

<xsl:if test="Marital_Status[.='Married']">
  <xsl:if test="NumberOfDependants[.>'2']">
    <xsl:if test="Income_Levels[.<'20000']">
      <Fuel_Consumption>interesting</Fuel_Consumption>
    </xsl:if>
  </xsl:if>
</xsl:if>

```



```

        </xsl:if>
    </xsl:if>
</xsl:if>

```

The meaning of this statement is as follows: **If Customer's Marital Status is "Married" and If the Number of Dependants is greater than 2 and if his Income Levels is less than £20000 then the Fuel Consumption of the car is something that he would like to see and consider.**

The second one is an **If Then Else Statement**.

```

<xsl:choose>
    <xsl:when test="NumberOfDependants[.>'2']">
        <bodydescription>interesting</bodydescription>
    </xsl:when>
    <xsl:otherwise>
        <Performance>not interesting</Performance>
    </xsl:otherwise>
</xsl:choose>

```

The meaning of this statement is as follows: **If the Number Of Dependants is greater than 2 then the body description of the car is something that he would like to see and consider. And also if this is the case he doesn't care about the performance of the car.**

The third one is an **If Then Else Statement**.

```

<xsl:choose>
    <xsl:when test="Sex[.='Male']">
        <Performance>interesting</Performance>
    </xsl:when>
    <xsl:otherwise>
        <Pasive_Safety>interesting</Pasive_Safety>
    </xsl:otherwise>
</xsl:choose>

```

The meaning of this statement is as follows: **If the Sex of the Customer is Male then the Performance of the Product is something that he would like to consider. Otherwise, which means that the Customer is Female, the Passive Safety of the car is something that she would like to consider.**

In figure 39 is the first customer's profile. By looking at the values of its elements we are able to see that because he is male, married, has three children and salary less than £2000 the product elements Fuel_Consumption, bodydescription, and performance are something that he wants to consider. In customized product view (shown in Figure 38) displayed whenever he selects a specific product we can see that only these product

elements have been selected (and their sub elements, if they contain any). Whenever the user makes a selection in the Product List Page (ListOfProducts.aspx) (Shown in Figure 37) he is redirected to the Page More Info Page (SCMoreInfoPage.aspx). Whenever this page is called a set of code is being executed (which actually resides “behind” this page):

```
namespace PhdShoppingBasket
{
    /// <summary>
    /// Summary description for SCMoreInfoPage.
    /// </summary>

    public class SCMoreInfoPage : System.Web.UI.Page
    {
        public short intProdID = 0;
        public short intXSLTRulesID = 0;
        public short intCustomerID = 0;
        public string BrandName = "";
        public string BodyStyle = "";
        public string EngineSize= "";

        private void Page_Load(object sender, System.EventArgs e)
        {
            // Put user code to initialize the page here
        }

        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            // CODEGEN: This call is required by the ASP.NET Web Form //Designer.

            InitializeComponent();
            base.OnInit(e);
            intProdID = short.Parse(Request.QueryString.Get("ProdID"));
            BrandName =Request.QueryString.Get("BrandName").ToString();
            BodyStyle =Request.QueryString.Get("BodyStyle").ToString();
            EngineSize=Request.QueryString.Get("EngineSize").ToString();
            intXSLTRulesID =
short.Parse(Session["XSLTRulesID"].ToString());
            intCustomerID = short.Parse(Session["CustomerID"].ToString());
            ExtractCustomizedProdInfo();
        }
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.Load += new System.EventHandler(this.Page_Load);
        }
        #endregion
        public void ExtractCustomizedProdInfo()
        {
            prjWrapper.clsWrapper Wrapper = new
prjWrapper.clsWrapperClass();
            Wrapper.InitializeVariables(intCustomerID,intProdID,intXSLTRulesID
```



```

D)
    Wrapper.CustomizedProductView(BrandName, EngineSize, BodyStyle);
    prjCleanUp.clsCleanUp CleanUp =
newprjCleanUp.clsCleanUpClass();
    CleanUp.ClenUp();
    Server.Transfer("CustomizedProdDetails.aspx");
    }
    }
}

```

When SCMoreInfoPage.aspx is loaded the code behind which resides in SCMoreInfoPage.cs is executed. The first member function to be executed is *override protected void OnInit(EventArgs e)*. Some of the member data such as intProdID, BrandName, BodyStyle, EngineSize, intXSLTRulesID, intCustomerID are initialized. Some of the data used to initialize these local variables are passed using a query string from the Product List. The code which actually passes these variables from the Product List page to the SCMoreInfoPage.cs is listed bellow:

```

<Ahref='SCMoreInfoPage.aspx?ProdID=<%#DataBinder.Eval(Container.DataItem,
"ProductIdentificationNumber")%>&amp;BrandName=<%#DataBinder.Eval(
Container.DataItem,"ProductName")%>&amp;EngineSize=<%#DataBinder.Eval(
Container.DataItem,"EngineSize")%>&amp;BodyStyle=<%#DataBinder.Eval(
Container.DataItem,"BodyStyle")%>'target=ShowMoreProdInfor><%#DataBind
er.Eval(Container.DataItem,"ProductName")%></A>

```

This code listed above is executed whenever a user makes a selection from the Product List (ListOfProducts.aspx). Once these information are passed from the Product List Page (ListOfProducts.aspx) to the Show More Info Page (SCMoreInfoPage.aspx) and the member variables are initialized through the execution of the OnInit member function the next thing that happens is the execution of the member function called **public void ExtractCustomizedProdInfo()**. This is where all of the processing happens and the desired information from a specific product is being represented to a specific customer.

The code which is executed then is listed below:

```

public void ExtractCustomizedProdInfo()
{
    prjWrapper.clsWrapper Wrapper = new
prjWrapper.clsWrapperClass();
    Wrapper.InitializeVariables(intCustomerID, ntProdID, ntXSLTRulesI
D)
    Wrapper.CustomizedProductView(BrandName, EngineSize, BodyStyle);

```



```

        prjCleanUp.clsCleanUp Cleanup =
newprjCleanUp.clsCleanUpClass();
        Cleanup.ClenUp();
        Server.Transfer("CustomizedProdDetails.aspx");
    }

```

In figure 33 where the overall architecture of the application is described we can see that SCMoreInfoPage.aspx communicates with the prjWrapper, which is actually a COM object, created using VB6, and uses its services, through RCW (Runtime Callable Wrapper). So the first thing that it happens is to create an instance, an object of the clsWrapperClass class which resides within the prjWrapper COM component. This is done by calling the code:

```

prjWrapper.clsWrapper Wrapper = new
prjWrapper.clsWrapperClass();

```

Once we have done this we can use this object called Wrapper and call some of its member functions. The first member function which is called is *InitializeVariables*. This is called in order to initialize some local variables to the COM Object.

```

Wrapper.InitializeVariables(intCustomerID,ntProdID,ntXSLTRulesID
)

```

The next member function which is called and performs the bulk of the customization operations is called *CustomizedProductView*.

```

Wrapper.CustomizedProductView(BrandName,EngineSize,BodyStyle);

```

The code for these two member functions is listed below (VB6 was used in order to create this COM object).

```

Public Sub InitializeVariables(ByVal intCustID As Integer, ByVal intProdID As
Integer, ByVal intXSLTRID As Integer)
    intCustomerID = intCustID
    intProductID = intProdID
    intXSLTRulesID = intXSLTRID
End Sub

```

The *InitializeVariables* member function initializes three local variables to the COM Object needed in order to extract the three main business documents involved in the personalization process namely Customer Profile identified by the CustomerId variable, Product XML File identified by intProductID variable, and XSLT Rules identified by intXSLTRulesID variable.

The next member function called *CustomizedProductView* does the bulk of the customization operations.

The code of this function is listed bellow.

```
Public Sub CustomizedProductView(ByVal strBrandName As String, ByVal
strEngineSize As String, ByVal strBodyStyle As String)
    Dim pausetime
    Dim start

    BrandName = strBrandName
    EngineSize = strEngineSize
    BodyStyle = strBodyStyle

    'Create the Customer XML File
    Set objCreateCustomerXML = New prjCreateCustXMLFile.clsCustXMLfile
    objCreateCustomerXML.CreateCustXMLFile intCustomerID

    'Create The Product XML File
    Set objCreateProductXML = New prjCreateProdXML.clsCreateProdXML
    objCreateProductXML.CreateProdXML intProductID

    'Extract The XSLT Rules
    Set objExtractXSLTRules = New prjExtractXSLT.clsExtractXSLT
    objExtractXSLTRules.ExtractXSLTRules intXSLTRulesID

    'Create The Intermediate Customer Profile
    Set objMyShell = New prjShell.clsActivateShell
    objMyShell.RunShellCom "C:/WorkingArea/saxon -o
C:/WorkingArea/XMLCustomizedCustProfile.xml
C:/WorkingArea/XMLCustomerProfile.xml C:/WorkingArea/XSLTRules.xsl"

    pausetime = 5
    start = Timer
    Do While Timer < start + pausetime

    Loop
    Set objCustProdXSLT = New prjECPI.clsECPI
    objCustProdXSLT.InitializePaths
"C:/WorkingArea/XMLCustomizedCustProfile.xml",
"C:/WorkingArea/XMLProduct.xml"
    objCustProdXSLT.GenerateXSLT
    pausetime = 5
    start = Timer
    Do While Timer < start + pausetime

    Loop
    'Create the customized view of the product for the specific customer
    objMyShell.RunShellCom "C:/WorkingArea/saxon -o C:/WorkingArea/Final.xml"
```


C:/WorkingArea/XMLProduct.xml C:/WorkingArea/CustomizedProductView.xsl"

```
pausetime = 5  
start = Timer  
Do While Timer < start + pausetime
```

```
Loop  
Set objProdCustView = New prjCustProdInfo.clsCProdInfo  
objProdCustView.CreateCustomizedProdDetails BrandName, BodyStyle,  
EngineSize  
End Sub
```

After some kind of initializations for some local variables this function performs the following:

a) Create the Customer XML File. The code which executes this one is listed

Bellow:

```
Set objCreateCustomerXML = New prjCreateCustXMLFile.clsCustXMLfile  
objCreateCustomerXML.CreateCustXMLFile intCustomerID
```

prjCreateCustXMLFile is a COM Component which has a member function called CreateCustXMLFile which accepts the intCustomerID as an argument and by employing the DOM (Document Object Model) creates the Customer XML File. DOM was used through the creation of XML files instead of employing the XML Option in SQL statements because it provides better flexibility in the creation of XML Files.

b) Create the Product XML File. The code which executes this one is listed

Bellow:

```
Set objCreateProductXML = New prjCreateProdXML.clsCreateProdXML  
objCreateProductXML.CreateProdXML intProductID
```

prjCreateProdXML is a COM Component which has a member function called CreateProdXML which accepts the intProductID as an argument and by employing the DOM (Document Object Model) creates the Product XML File. DOM was used through the creation of XML files instead of employing the XML Option in SQL statements because it provides better flexibility in the creation of XML Files.

c) Extracts The XSLT Rules. The code which performs this action is listed bellow:

```
Set objExtractXSLTRules = New prjExtractXSLT.clsExtractXSLT  
objExtractXSLTRules.ExtractXSLTRules intXSLTRulesID
```

prjExtractXSLT is a COM Component which has a member function called ExtractXSLTRules which accepts the intXSLTRulesID as an argument and by employing the DOM (Document Object Model) extracts the XSLT Rules file.

d) Create The Intermediate Customer Profile. Here I have created a wrapper around the command shell in order to run a XSLT Parser called SAXON.

```
Set objMyShell = New prjShell.clsActivateShell  
objMyShell.RunShellCom "C:/WorkingArea/saxon -o  
C:/WorkingArea/XMLCustomizedCustProfile.xml  
C:/WorkingArea/XMLCustomerProfile.xml C:/WorkingArea/XSLTRules.xsl"
```

The code of the RunShellCom is listed bellow:

```
Public Sub RunShellCom(ByVal ListOfArguments As String)  
    Shell ListOfArguments, vbHide  
End Sub
```

Which is translated whenever the ListOfArguments is populated to the following statement:

```
Shell "C:/WorkingArea/saxon -o C:/WorkingArea/XMLCustomizedCustProfile.xml  
C:/WorkingArea/XMLCustomerProfile.xml C:/WorkingArea/XSLTRules.xsl"
```

This is a way which is employed by VB applications to run an external executable. A command called Shell is employed followed by the name of the external executable (in this case the external executable is SAXON) and then SAXON accepts some parameters. The first parameter is the destination path of the resulting xml file called ****XMLCustomizedCustProfile.xml**** . The input parameters are the Customer Profile which has been retrieved recently and saved temporarily to the C:/WorkingArea folder (****C:/WorkingArea/XMLCustomerProfile.xml****) and the XSLT Rules stored also temporarily to the C:/WorkingArea folder (****C:/WorkingArea/XSLTRules.xsl****). The Customer Profile for the first Customer can be seen in Figure 39 and the XSLT Rules for the Customer Category to which our first customer belongs can be seen at Figure 41. The resulting file when we execute the Shell command called

XMLCustomizedCustProfile.xml and stored temporarily to folder C:/WorkingArea/ contains the following information:

```
<?xml version="1.0" encoding="utf-8" ?>
- <profile>
  <Fuel_Consumption>interesting</Fuel_Consumption>
  <bodydescription>interesting</bodydescription>
  <Performance>interesting</Performance>
</profile>
```

Figure 42: *XMLCustomizedCustProfile.xml* contents.

e) The next step is crucial in a sense that creates another XSLT file which is going to be used in order to extract all the relevant product information for the specific customer.

```
Set objCustProdXSLT = New prjECPI.clsECPI
objCustProdXSLT.InitializePaths
"C:/WorkingArea/XMLCustomizedCustProfile.xml",
"C:/WorkingArea/XMLProduct.xml"
objCustProdXSLT.GenerateXSLT
```

The objCustProdXSLT object which is an instance of the COM component prjECPI.clsECPI has a member variable called GenerateXSLT. This member function contains code which parses the specific product information and based on the information supplied from the *XMLCustomizedCustProfile.xml* file produces dynamically yet another XSLT file which is to be used in order to extract all the relevant information from the product xml file. So the input for this member function are two files: The *XMLCustomizedCustProfile.xml* which can be seen in Figure 42 and the product information which can be seen in Figure 40. The output of this process is another XSLT file called *CustomizedProductView.xsl* with contents which can be seen in Figure 43.


```

- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
- <xsl:template match="/">
- <Car>
-   <xsl:apply-templates select="/Car/features/Fuel_Consumption" />
-   <xsl:apply-templates select="/Car/features/bodydescription" />
-   <xsl:apply-templates select="/Car/features/Performance" />
- </Car>
- </xsl:template>
- <xsl:template match="Fuel_Consumption">
- <Fuel_Consumption>
- <Urban_Cycle>
-   <xsl:value-of select="Urban_Cycle" />
- </Urban_Cycle>
- <Highway_Cycle>
-   <xsl:value-of select="Highway_Cycle" />
- </Highway_Cycle>
- </Fuel_Consumption>
- </xsl:template>
- <xsl:template match="bodydescription">
- <bodydescription>
- <bodystyle>
-   <xsl:value-of select="bodystyle" />
- </bodystyle>
- <doors>
-   <xsl:value-of select="doors" />
- </doors>
- <bootcapabilities>
-   <xsl:value-of select="bootcapabilities" />
- </bootcapabilities>
- </bodydescription>
- </xsl:template>
- <xsl:template match="Performance">
- <Performance>
- <Maximum_Speed>
-   <xsl:value-of select="Maximum_Speed" />
- </Maximum_Speed>
- <Acceleration>
-   <xsl:value-of select="Acceleration" />
- </Acceleration>
- </Performance>
- </xsl:template>
- </xsl:stylesheet>

```

Figure 43: CustomizedProductView.xsl contents.

f) Create the customized view of the product for the specific customer.

```

objMyShell.RunShellCom "C:/WorkingArea/saxon -o C:/WorkingArea/Final.xml
C:/WorkingArea/XMLProduct.xml C:/WorkingArea/CustomizedProductView.xsl"

```

Here I have created a wrapper around the command shell in order to run a XSLT Parser called SAXON.

The code of the RunShellCom is listed bellow:

```

Public Sub RunShellCom(ByVal ListOfArguments As String)

```

```
Shell ListOfArguments, vbHide
```

```
End Sub
```

Which is translated whenever the ListOfArguments is populated to the following statement:

```
Shell "C:/WorkingArea/saxon -o C:/WorkingArea/Final.xml
```

```
C:/WorkingArea/XMLProduct.xml C:/WorkingArea/CustomizedProductView.xsl"
```

By executing this command a new file is created which is stored in C:/WorkingArea and is called Final.xml which actually contains only the relevant information for the specific customer. The contents of this file can be seen in figure 44.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Car>
  - <Fuel_Consumption>
    <Urban_Cycle>12 lt</Urban_Cycle>
    <Highway_Cycle>13.5 lt</Highway_Cycle>
  </Fuel_Consumption>
  - <bodydescription>
    <bodystyle>Station Wagon</bodystyle>
    <doors>4</doors>
    <bootcapabilities>700</bootcapabilities>
  </bodydescription>
  - <Performance>
    <Maximum_Speed>200</Maximum_Speed>
    <Acceleration>10.5</Acceleration>
  </Performance>
</Car>
```

Figure 44: Customized Product Information.(Final.xml)

g) The last code to be executed in the CustomizedProductView member function is the following:

```
Set objProdCustView = New prjCustProdInfo.clsCProdInfo
```

```
objProdCustView.CreateCustomizedProdDetails BrandName, BodyStyle, EngineSize
```

This code creates an html table and places all the data found in Final.xml file inside the table. Also create an aspx page in order to display the data. The file is create in the

following

path:

C:/Inetpub/wwwroot/PhdShoppingBasket/CustomizedProdDetails.aspx.

Then the last instruction which is executed is:

```
Server.Transfer("CustomizedProdDetails.aspx");
```

Which actually displays the contents of the customized product information.

Now it's time to create a second customer and register his data in the system. If we perform the steps described in figures 34 to 36 we have a new customer with the following details:

The screenshot shows a Microsoft Internet Explorer window titled "NewCustomerRegPage - Microsoft Internet Explorer". The address bar displays "http://localhost/PhdShoppingBasket/NewCustomerRegPage.aspx". The page content is titled "New Customer Registration Page" and contains a form with the following fields and values:

Customer ID	Customer2
Password	Password2
Marital Status	Single
Sex	Male
No. Of Dependants	0
Income	27000

Below the form is a "Save" button. The browser's status bar at the bottom shows "Done" and "Local intranet".

Figure 45: Second Customer registration information.

Whenever the customer selects the Save Command button is being redirected to the product list.

	Brand Name	Engine Size	Doors
1	<u>BMW</u>	1800	2
2	<u>Ford</u>	1600	4
3	<u>Mercedes</u>	2000	2
4	<u>Mazerati</u>	2500	2
5	<u>Mercedes</u>	2000	4

Figure 46: Product List for second customer currently registered in the system.

If the user now makes the same selection as the first user did, e.g. selection number 5 which is a Mercedes car is to be presented with the following customized view of the product. Taking into account the personal data entered by the second customer which can be seen at Figure 45, the XSLT Set of Rules associated with this customer which can be seen at Figure 32, we expect the customized product view for the second customer to be different than the first's one. And indeed this is the case. The Customized product information for the second customer can be seen in Figure 47.

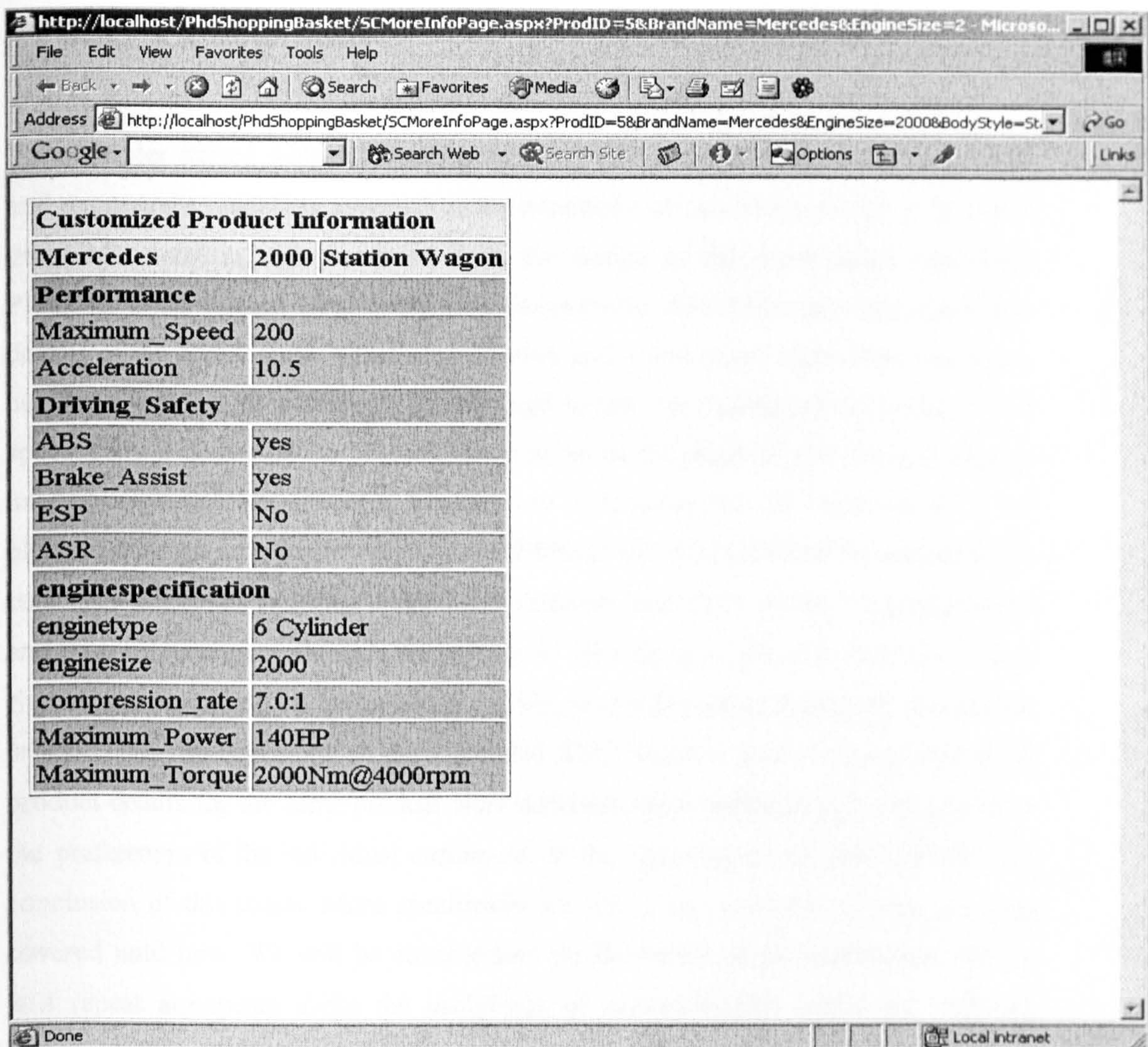


Figure 47: Customized product view for the second customer.

As we can see from Figures 38 and 47 the product details for the same product are different and are dependent on the different users selecting the product and their different profiles and their different XSLT Rules associated with these customers.

4.5 Conclusions

This Chapter started with an overview of the problem of designing software systems and suggested a modelling approach as the potential solution. Subsequently it discussed the UML notation and proceeded with the design of the Application Integration Platform using various UML modelling perspectives. Also presented implementation details of the XSLT Rules Creator application and a web based application, actually a cut down version, of a shopping basket used to test the customization process. More specifically it discussed the physical architecture of the platform and the technologies used to implement the platform. Furthermore it demonstrated the functionality of the platform through an example creating two different set of XSLT Rules for two different customer categories and then created two different customers for the shopping basket and then we have gone through the process of viewing some product details. We have shown that depending on the customer profile, as it was created during the registration process, and also depending on the associated XSLT Rules to each customer profile the product details for the same product were different, were customized in order to meet the preferences of the individual customers. In the next chapter we will represent the conclusion of this thesis. More specifically we will make a review of what we have covered until now. We will be concentrated on the notion of personalization and we will repeat arguments about the usefulness of personalization, about the different personalization processes available and about the problems associated with the different personalization techniques. We will revisit the thesis's objectives, the results which were achieved will be represented and some testing for the research hypothesis will be given. As a conclusion a section which discusses ideas about further research is included.

Chapter 5 Conclusions

5.1 Introduction

As business-to-customer electronic market has already become an established way of doing business on the web, competition in the online retail marketplace is growing. Profit margins are affected from this stiff competition from existing competitors and also from new companies entering the market. To endure in this environment and gain a competitive advantage, a successful online vendor must provide a collection of captivating, personalized services that meet its customer needs. Gradually e-commerce sites are investing more on personalization studies in order to solve this problem. Internet's expansion now days makes it more possible that most companies they will have their first interaction with a potential customer via the company's web site.

As a result, it is crucial for companies, to personalize their web sites, in order to match and transform web site content, e-mail and other communication means, used to interact with customers, to individual customer preferences. Companies seek to make potential customers increase the amount of time that they spend on their site, as well as increase their willingness to return to their site, which might lead to increased possibilities of purchasing a product. The company's objective is not to sell only a product or a service to a specific customer, but to increase the possibility that he/she is going to become a loyal customer and the company can eventually sell items and services to the specific customer over his/her lifetime. The ability to collect customer data exceeds our capability to analyze that data. And our capability to analyze customer data also exceeds our ability to act on that analysis while interacting with the customer. As a result, businesses lose the personal touch and push away their customers. Large amounts of unfiltered content and raw data are forbidding business from personalizing the customer experience, making customers to quit from interactions with the impression that the enterprise doesn't treat them as individuals. This leads to disappointment, lowers customer tendency towards loyalty and increases the probability that customers will try to find a different online retailer.

An obvious solution to this problem is the automation of the personalization process, i.e. the utilization of automated techniques that manipulate the vast masses of product and customer data.

Some of the most commonly used personalization techniques include:

1. **Content-based filtering**, This group of techniques applies machine learning methods to Web content, primarily text, in order to discover the personal preferences of a user. Content-based filtering works by analyzing the content of the objects to form a representation of the visitor's interests.
2. **Collaborative filtering** is a technology which looks at a user's behavior, relates it to the behavior of other users, and makes recommendations of other products or information that the user will care about. Collaborating filtering system makes recommendations to a specific user which are based on other "similar" users who have purchased products before.
3. **Rules-based personalization** uses complex business rules, which depict business practices in logical constructs, i.e. in the form of a conditional statement: If X and Y, then Z. Rule-based techniques can be used in conjunction with filtering techniques, in order to develop the best recommendation.

Problems associated to these techniques are as follows.

Collaborative filtering techniques suffer from:

- **Sparsity**. In real life conditions, many commercial recommender systems are employed in order to evaluate large product sets (e.g. Amazon.com makes among other things and recommendations about books. Also CDnow.com recommends music albums). In these kinds of systems even active and quite dedicated customers most probably have purchased less than a 0.5% of the available products (0.5% of 2,000,000 books is 10,000 books). As a result, a recommender system based on the usage of the nearest neighbour algorithms may not be able to make any recommendations at all for a particular user. This problem is known as reduced coverage. Also, the correctness of the produced recommendations may be poor. An example of a poor recommendation can be caused by the loss of neighbour transitivity. E.g. if customer A and B have a strong association and B also associates highly with C, it is not necessarily true that A and C will be associated as they may have bought very few common products.

- **Scalability.** Nearest neighbour algorithms performance is heavily related to both the number of customers and the number of products. When we run these algorithms in order to support a system which gives recommendations to millions of customers about millions of products, then the system will suffer serious scalability problems.
- **Cold-start problem.** One tricky, although common problem for a recommender system is the *cold-start* problem, where suggestions are needed for items that no one (in our data set) has yet rated. Pure collaborative filtering don't perform well on this situation, since no one has given any ratings for a product in order this rating to form a basis for recommendations.

Problems of using business rules:

- Personalization business rules can get very complicated. A rule engine is needed in order to manage the rules. A rule management system is needed in order to test and validate new rules before they are applied in production. Controls are needed in order to ensure that a new rule won't duplicate or conflict with an existing rule. Producing effective CRM personalization business rules is very difficult and challenging, and it requires both business knowledge and technical skills.
- Rules-based personalization implies and requires that a detailed knowledge of what the customer wants. The solutions are also not very scalable, because the business rules need to be constantly altered and changed in order to capture the customer needs. A well-defined process for changing the rules is needed. Also users can play a vital role by modifying the rules by providing explicit preferences and commands.
- Business rules cannot scale to handle millions of customers on an individual basis. It is simply too complicated. To simplify the challenge, business rules typically are applied to customers that are grouped into segments with similar behavior patterns. Because of this, business rules cannot handle true one to one marketing, and do not deliver on the true value of decision making at the customer level.

Problems of Content-based filtering include:

- **Content Limitation:** Information Retrieval methods can only be applied on a specific kind of content, such as text and image, and the extracted features can only capture certain aspects of the content. When a product consists of hundreds or even thousands of attributes then it is difficult for this approach to work efficiently, because content-based filtering is heavily based on previous user selection. It is most unlikely that a user will deal with this amount of information and therefore the feature recommendations which will be based on a partially finished selection process will be inaccurate.
- Content-based filtering is the simplest filtering approach that filters content by keywords, or string matching. The majority of Web search engines use content-based filtering, but they manage to collect only a small part of the indexable Web, less than 30 percent, according to [Lawrence & Lee Giles, 1998] and also users have to go through many results to find out relevant selections. According to [Florescu *et al*, 1998] low coverage is achieved because the majority of Web pages are dynamically created and therefore they are not directly accessible via hyperlinks. Another reason is the lack of well defined conceptual models for Web information retrieval.

5.2 Revisiting the thesis objectives

The main research objective of the project is to investigate how XSLT technologies can be used for the development of matching engines that find XML represented products that match the tastes, needs or requirements of customers as captured in customer profiles, also represented in XML. More specifically our research investigates into novel algorithms for transforming XML based product specifications using rules that derive from mining customer profiles with the purpose of customizing the product information.

Other research objectives are:

- Research into techniques for constructing and representing customer profiles. Currently customer profiles are constructed either implicitly or explicitly. In explicit profiling every customer is asked to fill out information or online forms. This method has an advantage because it is allowing customers to describe to the site what they want to see. Implicit profiling monitors the visitor's behavior. The technique is generally

not obvious to the visitor. Browsing and buying patterns are the behaviors most often measured. The browsing pattern is monitored by using cookies. There are some problems though, associated to the construction and maintenance of customer profiles. Data reliability is one problem. Many customers are being afraid to pass personal information to a Web site. The problem is how to produce more accurate profiles which will represent the reality while it is known that data given by customers are not real or they are partially true. Maintainability is also another problem. Enterprises establish more than one communication means (e.g. mail, telephone, text messaging) in order to communicate with their potential customers. Data gathered by using these communication means which play role in the construction of user's profile are stored by using different formats in different storage media. Keeping all these customer data synchronized and credible is a complex task.

- Research into concepts and techniques for product schemas. Product schemas describe the product features (properties/attributes). XML is used in order to create the product schemas. Problem with the change of product specification which has to be reflected back to the product schemas. Complex products can be represented by XML notation since nested XML document elements can be combined and nested in order to build complex information structure. But in most of real world situation product specifications are stored in a relational table in a database. In principle, it is possible to convert complex XML documents for use with relational technology, but this tends to be inefficient. XML documents by definition are hierarchically structured. Nested XML document elements can be combined and nested in order to build complex information structures. In principle, it is possible to convert complex XML documents for use with relational technology, but this tends to be inefficient. Mapping XML elements to and from relational tables is a slow process and requires normalization and optimization effort before a XML document can be stored properly.
- Construct and validate algorithms for personalizing XML instances of product information using XSLT rules that match customer profiles to products. The research problem is to identify and design all possible types of transformations that can be affected upon a product schema. XML is used for the construction of the product and customer meta data files. Personalization process involved a way of transforming these meta data files. Another W3C standard was employed in order to argue, create rules and relationships between the products and the customers. XSLT is a flexible language, which includes the ability of performing complex computational tasks by

incorporating scripts written in a procedural language. There is some performance consideration related to the number of instruction of the XSLT Meta data descriptor file. Less complex XSLT instructions reduce the execution time of the XSLT engines (the parser which evaluates and then executes a set of XSLT instructions).

5.3 Results Achieved

This research has culminated in a hybrid personalization approach that combines elements from different, well known and established personalization techniques. Namely, it uses the concept of rules, as used by Rules-Based personalization systems. Its novelty though is that the rules are defined using a new standard (XSLT) and they are obtained by a data mining tool. Our approach shares also commonalities with collaborative systems, because such systems use extensively data mining tools, for knowledge extraction. The prototype is a platform which uses the functionality of different components in order to perform the personalization process. It's expandable in a sense that:

1. Its components are loosely coupled
2. The interaction between them is done by using Meta Data descriptor files (hierarchical text files that can be either XML or XSLT files). So, no extra logic has to be implemented for the integration of the information passed to deferent components. This is possible because a programming model, called DOM (Document object Model), is employed which facilitates the parsing and information extraction from Meta Data descriptor files.
3. New components providing enhanced business functionality can be easily added since the interfacing between them is done by a well defined way.

Metadata usage. In our approach the XML standard is being employed in order to describe the customer profile-s and to represent the structure of products in a web site. Product/service information can be described in a neutral and platform independent format (XML) and delivered to the customer over multiple communication media and technologies. Ideally the customer can receive customized product/service information directly to the device of his choice, e.g. a Web browser, mobile phone, digital TV etc.

Meta data descriptor files associate different entities. The XSLT [W3C 1999] standard is adopted for the creation of the association. Because of the Meta data file nature, their

structure can be represented as a tree. There is an already established technique of manipulating such kind of trees called DOM [W3C 2004]. Although rule-based personalization techniques suffer because the business rules need to be constantly altered and changed in order to capture the customer needs, the introduction of XML and XSLT and the usage of DOM (Document Object Model) make the task of alteration and synchronization of business rules embedded in an XSLT file an easier task.

5.4 Research hypotheses validation

To test the performance and scalability of the proposed system we designed and run a number of tests. The test environment consisted of a PC with a 3.06 GHz Intel Zeon processor and 1 Gigabyte of RAM running Microsoft 2000 Professional Edition. The SAXON XSLT processor was used.

RESPONSE TIME (SECS)	NUMBER OF ELEMENTS IN XML SCHEMAS	SIZE OF XML FILE	NUMBER OF RULES
4.0	30	50kb	5
5.2	50	100kb	8
7.4	90	200kb	14
12.8	180	400kb	22
26.1	360	800kb	35
49.7	500	1600kb	67

Table 1: Test results for various configurations.

Several tests involving different sizes of XML Schemas and number of rules were run. The table 1 above shows the performance of the system for different sizes of data. The results prove that the system is scalable, i.e. the response time is increasing almost linearly with the size of the XML Schemas and the number of XSLT rules.

PROBLEMS	SCARCITY		RULE	CONTENT BASED
----------	----------	--	------	------------------

Personalisation Approaches	OF DATA	SCALABILITY	MANAGEMENT	LIMITATIONS
Content-based filtering	Not affected	Affected	Not affected	Affected
Collaborative filtering	Affected	Affected	Not affected	Affected
Rule-based filtering	Not affected	Affected	Affected	Affected
OUR APPROACH	Not affected	Not Affected	Not affected	Not affected

Table 2: Comparison of different personalization techniques

We have created a table in order to compare the different approaches. The comparison is based on well known problems of the established personalization techniques and on the observation and experiment from the one proposed one. The problem of the scarcity of data doesn't affect the content-based filtering method because this technique doesn't employ any kind of search on an incomplete matrix. Collaborating filtering is affected negatively because is a technique which employs the nearest neighbour algorithm, and since the neighbour is not densely populated we have wrong estimations. Rule-based technique isn't affected because of the fact that employs a set of rules for the personalization process and doesn't search matrices. Our approach also doesn't affected because is it uses meta data descriptor files implemented as tree structures and the only thing that is interested on is the association rules residing in the XSLT rules meta data file.

The problem of scalability affects the content-based technique because they manage to collect only a small part of the indexable Web, less than 30 percent, according to [Lawrence & Lee Giles, 1998] and also users have to go through many results to find out relevant selections. According to [Florescu *et all*, 1998] low coverage is achieved because the majority of Web pages are dynamically created and therefore they are not directly accessible via hyperlinks. So whenever we have bigger workload and more data to process the failure percentage is also high. As far as it concerns collaborative filtering then we can say that is badly affected because of the fact that the time complexity of executing the nearest-neighbour algorithm grows linearly with the number of items and the number of users. Thus, the recommendation system cannot support large-scale applications with an efficient way because it takes to much time in order to traverse the matrix. Rule based technique doesn't perform well when it comes

to handle large amounts of business rules, some of them which can be quite complex, and they might be changing so rapidly. Personalization business rules can get very complicated. A rule engine or a rule-base is needed in order to manage the rules. A rule management system is needed in order to test and validate new rules before they are applied in production. Controls are needed in order to ensure that a new rule won't duplicate or conflict with an existing rule. Producing effective CRM personalization business rules is very difficult and challenging, and it requires both business knowledge and technical skills. We assume rule derivation from a data mining engine. The structure is XSLT, which is tree structure and there is a well defined way to treat and enhance and manipulate this tree by the use of DOM (Document Object Model). Also some benchmark results shows that the engine exhibits almost linear behavior. This actually means that large amounts of data can be manipulated and the amount of time spent is to be within logical and acceptable margins.

Rule management affects negatively the rule-based technique. The reason here is similar to the one described in the previous section. Personalization business rules can get very complicated. A lot of effort is required in order to maintain these rules active and able to reflect customer's needs. And this is because many rule-based systems use a text editor to edit the business rules and their structure isn't easily modified afterwards. Our approach uses Meta data in order to visualize our business rules which are represented by tree structures. Also their manipulation is far simpler than the rule-based system's rules by exploiting DOM. Content-based and collaborative filtering techniques are not affected by the rule management factor because they don't use rules in order to perform personalization.

Content based limitations. Information Retrieval methods can only be applied on a specific kind of content, such as text and image, and the extracted features can only capture certain aspects of the content. When a product consists of hundreds or even thousands of attributes then it is difficult for this approach to work efficiently, because content-based filtering is heavily based on previous user selection. Collaborative filtering also suffers from the same limitation as not enough users (or no users at all) may have selected some of the product features.

5.5 Further Research

Future research can focus on enhancing the scalability and manageability of the proposed approach and facilitate its integration with other systems. This is further discussed below.

XML documents by definition are hierarchically structured. Nested XML document elements can be combined and nested in order to build complex information structures. In principle, it is possible to convert complex XML documents for use with relational technology, but this tends to be inefficient. Mapping XML elements to and from relational tables is a slow process and requires normalization and optimization effort before a XML document can be stored properly. Research can be conducted on how to create a faster way of mapping XML data to and from relational database. Until now we have used many techniques, employing DOM in order to extract the raw data from the data base and transform them to XML format.

Although the system is build by using a modular fashion, and its architecture is component based, we don't have any standard interface to be used in order to connect the system with a legacy system.

Another potential area of research can be the automated update of the XSLT rules based on changes of the constituent parts, e.g. customer profile Meta data file and the product Meta data file. This again is related with the problems of how to have a tighter association with data stored in a relational database system and their presentation in a user interface, and how to make the change of customer of product data in the data base to be reflected in their XML presentation.

5.6 Concluding Remarks

In conclusion, this thesis has produced a significant contribution to the problem of effective personalization of product information on the web. By combining XML schemas for customer profiles the product specifications with XSLT for designing personalization rules this approach overcomes many of the problems that characterize conventional personalization approaches such as lack of scalability and management of schema evolution.

REFERENCES

- [1] Maurino A., Fraternali P., *Commercial Tools for the Development of Personalized Web Applications: A Survey*, In E-Commerce and Web Technologies: Third International Conference, EC-Web 2002, Aix-en-Provence, France, September 2-6, 2002. Proceedings Editors: K. Bauknecht, A. Min Tjoa, G. Quirchmayr
- [2] Balabanović, M., Shoham, Y., *Fab: content-based, collaborative recommendation*, Communications of the ACM, Vol 40, No. 3, March 1997
- [3] Bernett Howard G., Kuhn Marcy D., *The Emergence of Electronic Customer Relationship Management*, The Telecommunications Review, Vol. 13, 2002.
- [4] CRML Available at: www.oasis-open.org
- [5] Cooley, R., Mobasher, B., and Srivastava, J., *Data preparation for mining world-wide Web browsing patterns*, Knowledge and Information Systems. Volume 1, Number 1, February 1999a
- [6] Cooley, R., TAN, P.-N., and Srivastava, J. *WebSIFT: The web site information filter system*, In Proceedings of the Web Usage Analysis and User Profiling Workshop (WEBKDD'99), Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999b.
- [7] Eirinaki M., Lampos H., Vazirgiannis M., Varlamis I., *SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process*, in Proceedings of SIGKDD '03,, Washington DC, USA, August 2003
- [8] Eirinaki M., Vazirgiannis M., *Web Mining for Web Personalization*, ACM Transactions on Internet Technology, Vol. 3, No. 1, Pages 1–27, February 2003.
- [9] Elofson, G. and Robinson, W.N.: *Creating a custom mass-production channel on the Internet*, in: Communications of the ACM, 41 (1998) 3, pp. 56-62.
- [10] Ester M., Gundlach S., Kriegel H.P., Sander J., *Database Primitives for Spatial Data Mining*, Institute for Computer Science, University of Munich
Proc. Int. Conf. on Databases in Office, Engineering and Science, (BTW '99), Germany, 1999

- [11] Fayerman M., *New Directions for Institutional Research*, Customer Relationship Management (Journal), Volume 2002, Issue 113, Pages: 57-68, 2002
- [12] Florescu D., A. Levy, and A. Mendelzon, *Database Techniques for the World Wide Web: A Survey*, *SIGMOD Record*, vol. 27, no. 3, pp. 59-74, Sept. 1998.
- [13] Forrester Research, *Firms Must Manage CRM Technologies to Keep Customer Service Costs from Skyrocketing*, 2001
- [14] Sonja Grabner-Kraeuter, Gernot Moedritscher, *Alternative approaches toward measuring CRM Performance*, 6th Research Conference on Relationship Marketing and Customer Relationship Management, Atlanta, June 9 – 12, 2002
- [15] Greenberg, P. *Capturing and Keeping Customers in Internet Real Time*, McGraw-Hill ISBN 0072127821, 2001
- [16] Hannigan, T., Palendrano, C., *Personalization Can Be Quite Dynamic*, DM Review, Feature Article in October 2002 Issue
- [17] Hijikata Y., *Implicit user profiling for on demand relevance feedback*, In International Conference on Intelligent User Interfaces , Proceedings of the 9th international conference on Intelligent user, Funchal, Madeira, Portugal, Pages: 198 – 205, ISBN 1-58113-815-6, 2004
- [18] Huang, Z., Chen, H., Zeng, D., *Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering*, ACM Transactions on Information Systems (TOIS), Volume 22, issue 1, 116-142, 2004
- [19] Imhoff, C., Loftis, L.; Geiger, J.G., *Building the Customer-Centric Enterprise: Data Warehousing Techniques for Supporting Customer Relationship Management*, Wiley ISBN 0471319813 London, 2001
- [20] Kirtland, M., *Designing Component-Based Applications*, Microsoft Press, ISBN 0735605238, 1999

- [21] KOBASA, A., *Personalized Hypermedia and International Privacy*, Communications of the ACM, Special Issue on Adaptive Web-based Systems and Adaptive Hypermedia, 64-67, 2002
- [22] Kosala, R., Blockeel, H., *Web mining research: A survey*. SIGKDD Explorations, 2000
- [23] Krulwich, B. and Burkey, C., *The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction*, IEEE Expert, September/October 1997.
- [24] Lawrence, S., and C. Lee Giles, *Searching the World Wide Web*, Science, vol. 280, no. 5,360, pp. 98-100, 1998
- [25] Lang, K., *Newsweeder: Learning to filter netnews*, In Proceedings of the 12th International Conference on Machine Learning (Tahoe City, Calif.) 1995
- [26] Lieberman, H., *Letizia: An agent that assists web browsing*, In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pages 924-929, 1995
- [27] Massegia, F., Poncelet, P., and Cicchetti, R., *WebTool: An integrated framework for data mining*, In Proceedings of the Ninth International Conference on Database and Expert Systems Applications (DEXA'99) (Florence, Italy, August), 892-901, 1999a
- [28] Massegia, F., Poncelet, P., and Teisseire, M., *Using data mining techniques on web access logs to dynamically improve hypertext structure*. In ACM SigWeb Lett., 8, 3 (Oct.) 13-19, 1999b
- [29] Mitchell M. Tseng and Frank Piller, *The Customer Centric Enterprise Advantages In Mass Customization and Personalization*, Springer ISBN 3-540-02492-1, 2003
- [30] Mobasher, B., Cooley, R., and Srivastava, J., *Creating adaptive web sites through usage based clustering of URLs*, In Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99), 1999
- [31] Mobasher, B., Cooley, R., Srivastava, J., *Automatic Personalization Based On Web Usage Mining*, August 2000, Vol. 43, No.8 Communications of the ACM, 142-151, 2000a
- [32] Mobasher, B., Dai, H., Luo, T., Sung, Y., and Zhu, J., *Discovery of aggregate usage profiles for web personalization*, In Proceedings of the Web Mining for E-Commerce Workshop

(WEBKDD'2000), Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, August, 2000b

[33] Mobasher, B., Dai, H., Luo, T., Sung, Y., and Zhu, J., *Integrating web usage and content mining for more effective personalization*, In Proceedings of the International Conference on Ecommerce and Web Technologies (ECWeb2000), Greenwich, UK, Sept., 2000c

[34] Mobasher, B., Dai, H., Luo, T., Nakagawa, M., Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. In workshop Intelligent Techniques for Web Personalization, editors S. Anand, B. Mobasher, pp 53-60, Seattle Washington, 2001

[35] Mooney J. R., Roy, L., *Content-based Book Recommending Using Learning for Text Categorization*, Proc. of the 5th ACM Conf. on Digital Libraries.

International Conference on Digital Libraries
San Antonio, Texas, United States

Pages: 195 - 204

Year of Publication: 2000

ISBN:1-58113-231-X

[36] Morita, M., Shinoda, Y., *Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval*, Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 272–281, 1994

[37] M.D. Mulvenna, S.S. Anand, A.G. Buchner, *Personalization on the Net using Web Mining*, Communications of the ACM, August 2000/Vol. 43, No. 8, 123-125

[38] Neuhold, E-J., *Personalization and User profiling & Recommender Systems*, Department of Computer Science and Business, University of Vienna, 2003

[39] NISS, National Institute of Statistical Sciences, Workshop on Data Quality, Technical Report Number 117, March, 2001

[40] Pazzani. M., *A Framework for Collaborative, Content-Based and Demographic Filtering*, Artificial Intelligence Review, Volume 13, Issue 5-6, 393 - 408, 1999

[41] PDX Available at: <http://xml.coverpages.org/pdx.html>

- [42] Peppers, D. and Rogers, M., *Enterprise one to one: Tools for competing in the interactive age*, New York 1997.
- [43] Pierrakos, D., Paliouras. G., Papatheodorou. C., Spyropoulos. C., *Web Usage Mining as a tool for Personalization: a survey*, User Modeling and User-Adapted Interaction, Volume 13, 311-372, 2003
- [44] Ramakrishnan, N., *PIPE: Web Personalization by Partial Evaluation*, IEEE Internet Computing, November/December 2000 (Vol. 4, No. 6), pp. 21-31
- [45] Renneberg, V., Borghoff, U-M., *Recommendations in Product Personalization: An Extended Example of Pipelined Filter Combination*, in: Proceedings of the 2nd Interdisciplinary World Congress on Mass Customization and Personalization (MCPC), 2003
- [46] Resnick, P. and Varian, H.: *Recommender systems*, in: Communications of the ACM, 40 (1997) 3, pp. 56-58.
- [47] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., *GroupLens, An Open Architecture for Collaborative Filtering of Netnews*. In Proceedings of ACM conference on Computer-Supported Cooperative Work, 175-186, 1994
- [48] Sakagami, H., Kamba, T., *Learning Personal Preferences on Online Newspaper Articles from User Behaviors*, Proc. of the Sixth International World Wide Web Conference, In Computer Networks and ISDN Systems, Vol. 29, pp. 1447–1456, 1997
- [49] Sarwar, B., Karypis, G., Konstan, J., Riedl J., *Analysis of recommendation algorithms for e-commerce*, Proceedings of the 2nd ACM conference on Electronic commerce, Minneapolis, Minnesota, United States, 158 - 167 , 2000
- [50] Schafer, B., J., Konstan, J., Riedi, J., *Recommender systems in e-commerce*, Electronic Commerce, Proceedings of the 1st ACM conference on Electronic commerce
Denver, Colorado, United States
Pages: 158 - 166
Year of Publication: 1999
ISBN:1-58113-176-3

- [51] Shahabi, C., Chen Y., *Web Information Personalization: Challenges and Approaches*, The 3rd International Workshop on Databases in Networked Information Systems (DNIS 2003), Japan, 2003
- [52] Smyth, B., Cotter, P., *A personalized television listings service*, Communications of the ACM, Volume 43, Issue 8, 107 – 111, 2000
- [53] Srivastava, J., Cooley, R., Deshpande, M., Tan P-N., *Web usage mining: discovery and applications of usage patterns from Web data*, ACM SIGKDD Explorations Newsletter Volume 1, Issue 2 (January 2000) Pages: 12 – 23, 2000
- [54] Shahabi, C., Zarkesh A.M., Adibi J., Shah V., *Knowledge discovery from users Web-page navigation*, Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications Page: 20, Year of Publication 1997, ISBN 0818678496
- [55] STEP Available at: <http://www.stepml.org/index.html>
- [56] Sweiger M., *Clickstream Data Warehousing*, John Wiley & Sons ISBN 0471083771, 2002
- [57] Schubert, P., *Persoanalization of E-Commerce Applications in SMEs: Conclusions from an empirical Study in Switzerland*, Journal of Electronic Commerce in Organizations, S. 22-40, July-Sept 2004
- [58] Terveen et al., *"PHOAKS: A System for Sharing Recommendations,"* Comm. ACM, vol. 40, no. 3, pp. 59-62, Mar. 1997.
- [59] Vandermerwe, S., *The Eleventh Commandment: Transforming to "Own" Customers*, John Wiley & Sons ISBN 0471958239, 1997
- [60] [W3C 2001] W3C XML Schema Part 0: Primer W3C Recommendation, 2 May 2001
- [61] [W3C 1999] W3C XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November 1999.
- [62] [W3C 2004] Document Object Model (DOM) Level 3 Core Specification Version 1.0 W3C Recommendation 07 April 2004

- [63] Zeng, C., Xing, C., Zhou, L., Zheng, X., *Similarity Measure and Instance Selection for Collaborative Filtering*, International Journal of Electronic Commerce, Volume 8, No. 4, 115 - 122, 2004